ALGEBRAIC APPROACHES FOR CONSTRUCTING MULTI-D WAVELETS

by

Fang Zheng

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

March, 2014

Abstract

Wavelets have been a powerful tool in data representation and had a growing impact on various signal processing applications. As multi-dimensional (multi-D) wavelets are needed in multi-D data representation, the construction methods of multi-D wavelets are of great interest. Tensor product has been the most prevailing method in multi-D wavelet construction, however, there are many limitations of tensor product that make it insufficient in some cases. In this dissertation, we provide three non-tensor-based methods to construct multi-D wavelets. The first method is an alternative to tensor product, called coset sum, to construct multi-D wavelets from a pair of 1-D biorthogonal refinement masks. Coset sum shares many important features of tensor product. It is associated with fast algorithms, which in certain cases, are faster than the tensor product fast algorithms. Moreover, it shows great potentials in image processing applications. The second method is a generalization of coset sum to non-dyadic dilation cases. In particular, we deal with the situations when the dilation matrix is $\Lambda = p\mathtt{I}_n$, where p is a prime number and $\mathtt{I}_n$ is the n-D identity matrix, thus we call it the prime coset sum method. Prime coset sum inherits many

ABSTRACT

advantages from coset sum including that it is also associated with fast algorithms. The third method is a relatively more general recipe to construct multi-D wavelets. Different from the first two methods, we attempt to solve the wavelet construction problem as a matrix equation problem. By employing the Quillen-Suslin Theorem in Algebraic Geometry, we are able to build n-D wavelets from a single n-D refinement mask. This method is more general in the sense that it works for any dilation matrix and does not assume additional constraints on the refinement masks.

This dissertation also includes one appendix on the topic of constructing directional wavelet filter banks.

Primary Reader: Youngmi Hur

Secondary Reader: Trac D. Tran

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Professor Youngmi Hur, who has been a tremendous mentor for me. I have learned a great deal from her unique perspective on research, her sharp insight on academic career, and her personal integrity and expectations of excellence. She has always been patient and encouraging to me, even when I do not have faith in myself. It is her enthusiasm and professionalism that influence me and lead me through this journey. Thanks to her unreservedly sharing the experience in and outside the academic world. I could not have achieved this far without her continuous support and motivation during my Ph.D study.

I would also like to thank my committee members, professor Carey Priebe, professor Trac Tran, and professor Sang Chin for their insightful comments and suggestions on my thesis. Their attendances made my defense a truly enjoyable moment.

Finally, a special thanks goes to my beloved parents. They always believe in me and give me the freedom to explore my own world. Their unconditional love and support are the most important things to me in the world.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Over the last couple of decades, wavelets have been one of the most popular and effective tools in data representation, therefore, have been used in various application areas such as signal processing and image processing. While constructing 1-D wavelets is well understood by now, constructing multi-dimensional (multi-D) wavelets is still a highly non-trivial problem. Since multi-D wavelet systems can be obtained from multi-D wavelet filter banks under some simple conditions [1–3], our main focus is laid on how to construct multi-D wavelet filter banks. There are two major problems in constructing multi-D wavelet filter banks. The first is to find a pair of multi-D biorthogonal lowpass filters. The second is to find the corresponding multi-D highpass filters such that they form a wavelet filter bank that satisfies the perfect reconstruction condition.

There has been great interest in both problems and a lot of researches have been done. The most well-known systematical way to construct multi-D wavelets is the *tensor product*, or so called *separable* wavelet construction method. An n-D 2^n-channel wavelet

filter bank is obtained by mapping a 1-D 2-channel wavelet filter bank through the tensor product operator. Due to its simplicity and generality, in practice, tensor product has been the prevailing method to construct multi-D wavelets. However, there are many limitations of the tensor product method, for example, i) the resulting multi-D highpass filters have directional preferences only along the coordinate directions, ii) the resulting multi-D lowpass and highpass filters have dense supports, which may result in slower implementation in real time, especially with high dimensions. Moreover, it only works for dyadic dilation wavelets, i.e., when the dilation matrix is $\Lambda = 2\mathtt{I}_n$, where $\mathtt{I}_n$ is the identity matrix. When non-dyadic frequency division is desired, tensor product method may be insufficient. Other non-tensor-based or non-separable wavelet construction methods have also been proposed in numerous literatures (see Section §2.1, §3.1.1 and §4.1 for the detailed discussion and the references therein). However, most of these methods only work for low dimensions (2-D or 3-D) or assume the lowpass filters satisfy certain additional conditions.

In this thesis, we attempt to tackle the two problems in multi-D wavelet construction through more generic way. We contribute three methods to construct non-separable multi-D non-redundant wavelet filter banks.

In Chapter §2, we present an alternative method to tensor product, called *coset sum*, to construct multi-D wavelets from a pair of 1-D biorthogonal refinement masks. The coset sum shares many essential features of the tensor product that make it attractive in practice: i) it preserves the biorthogonality of 1-D refinement masks, ii) it preserves the accuracy number of the 1-D refinement mask, and iii) the wavelet system associated with it has fast algorithms for computing and inverting the wavelet coefficients. The coset

sum overcomes the drawbacks of tensor product in the following way: i) the directional preferences of the multi-D highpass filters can be custom designed by initial input, due to which coset sum wavelet systems show promising results when applied to images with strong directional content (see Section §2.4.3 for some experimental results), ii) the coset sum multi-D filters have sparser supports than those of tensor product, due to which and other reasons, coset sum wavelet systems are associated with fast algorithms, whose *complexity constant* (cf. Section §2.4.2 for the definition of complexity constant) is independent of the spatial dimension n. It is well known that tensor product fast algorithms have a complexity constant that increases linearly with the spatial dimension n, thus in high dimension, coset sum fast algorithms have the potential to be much faster than tensor product fast algorithms. The coset sum method works for any spatial dimension, however, it only works for dyadic dilation wavelets, and needs one of the starting 1-D biorthogonal refinement masks to be interpolatory.

In Chapter §3, we generalize the coset sum method to include the situation when the dilation matrix is in the form of $\Lambda = p\mathtt{I}_n$, where p is a prime number. We call this the *prime coset sum* method. The prime coset sum method is a systematic way to construct multi-D wavelet filter banks with dilation matrix $\Lambda = p\mathtt{I}_n$ from two 1-D lowpass filters with dilation p, with the following attributes: i) the vanishing moments of the multi-D wavelet filter banks can be controlled by choosing proper 1-D lowpass filters with certain properties, and ii) similar to coset sum, the wavelet systems constructed by prime coset sum are associated with fast algorithms, whose complexity constant is independent of the spatial dimension n. Similar to coset sum, prime coset sum also needs one of the starting

1-D lowpass filters to be interpolatory.

For the above two methods, we both use an operator to map a pair of 1-D lowpass filters to a pair of multi-D lowpass filters first, then find a particular set of multi-D highpass filters such that they form wavelet filter banks. However, if we start with one single multi-D lowpass filter, and want to construct a multi-D wavelet filter bank from it, or additionally with the condition that the vanishing moments of the wavelet filter bank are at least as much as the accuracy number of the initial multi-D lowpass filter, the above two methods can not provide a solution. On the other hand, both coset sum and prime coset sum method assume the dilation matrix taking specific forms ($\Lambda = 2\mathtt{I}_n$ in coset sum case and $\Lambda = p\mathtt{I}_n$ in prime coset sum case), and require one of the initial 1-D lowpass filters to be interpolatory. To give a more general construction method, we take a different perspective. We consider the wavelet construction problem as solving matrix identities with Laurent polynomial entries.

In Chapter §4, we propose a new algebraic approach for constructing multi-D wavelet filter bank using the Quillen-Suslin Theorem for Laurent polynomials. Quillen-Suslin Theorem is used to transform the filters in polyphase representation to a special form of generalized polyphase representations, for which the existing matrix techniques can be readily applied. Our construction method presents some advantages over the traditional methods in the following ways: i) it works for any dimension and for any dilation matrix, ii) it does not require the initial lowpass filters to satisfy any additional assumption such as interpolatory condition, and (iii) it provides an algorithm for constructing n-D wavelet filter banks from a single n-D lowpass filter so that their vanishing moments are at least as many as the accuracy number of the lowpass filter.

In this dissertation, we also include one appendix on the topic of constructing directional wavelet filter banks. A method of using 1-D Neville filters to design non-redundant directional wavelet filter banks is discussed in the Appendix §5.

The work presented in Chapter §2 has been published in [4]. The work presented in Chapter §3 is in preparation for a separate research article. The work in Chapter §4 has been submitted as a research paper, and the work in Appendix §5 has been published separately in [5].

Chapter 2

An Alternative to Tensor Product in Multi-D Wavelet Construction: Coset Sum

2.1 Introduction

One of the most common tools for constructing wavelets is Multiresolution Analysis (MRA) [1]. In MRA, a multivariate biorthogonal wavelet system can be obtained from a pair of multivariate biorthogonal refinement masks. The tensor product has been the prevailing method for deriving a pair of multivariate biorthogonal refinement masks from a pair of biorthogonal univariate refinement masks.

In this paper we are interested in studying the operators that map lower dimensional refinement masks to higher dimensional refinement masks. Throughout this paper,

the multidimensional (multi-D) refinement masks that can be decomposed into lower dimensional refinement masks by such operators are referred to as *decomposable*. One such operator is the tensor product. The multi-D refinement masks obtained via tensor product are called *tensor product (or separable) refinement masks*. Since the word "separable" is reserved for the tensor product by the definition in the literature, we use the word "decomposable" to indicate more general case than the tensor product. It should be noted that a "nonseparable" refinement mask only means it is not a tensor product refinement mask, and it can still be a "decomposable" refinement mask. Tensor product can also be used to construct multi-D wavelet masks, which are called *tensor product (or separable) wavelet masks* (cf. Section §2.2.2).

In MRA setup, construction of multi-D biorthogonal wavelet systems can be done by two steps: (i) construction of multi-D biorthogonal refinement masks (or refinable functions); (ii) construction of multi-D wavelet masks. To construct a nonseparable multi-D wavelet system, one can try making the refinement masks nonseparable in step (i) or making wavelet masks nonseparable in step (ii). Since, once a pair of multivariate biorthogonal refinement masks are given, the matrix extension problem of finding wavelet masks can always be solved by using Quillen-Suslin theorem (see, for example, [6]), the main effort so far for constructing nonseparable wavelets has been made in step (i). However, we note that Quillen-Suslin theorem serves only as a guide since in the process of determining the wavelet masks, some parameters still need to be specified.

Although there have been many methods for constructing nonseparable multi-D wavelets [7–20], constructing nonseparable multi-D wavelet systems is highly nontrivial.

Many of these methods work only for low spatial dimensions (2-D or 3-D) and they cannot be easily extended to other dimensions. Others assume that the wavelets or refinable functions have a special form (e.g. the refinable function has a box spline factor) and cannot be easily generalized to other cases.

One of the disadvantages of the above approaches for constructing nonseparable wavelets is that they construct a pair of multi-D biorthogonal refinement masks essentially from scratch, which can be quite complicated, especially for high spatial dimensions. A simpler way to obtain multi-D biorthogonal refinement masks is to use an operator that maps 1-D biorthogonal refinement masks to multi-D biorthogonal decomposable refinement masks. Most of the existing nonseparable wavelet construction methods (e.g. [21–27]) that use decomposable refinement masks employ operators such as the McClellan transform for quincunx or other 2-channel sampling lattices.

Most multi-D wavelet systems that are used in practice nowadays are separable wavelet systems constructed by the tensor product of 1-D wavelet systems. In Section §2.2.2 we briefly discuss the use of tensor product in constructing biorthogonal wavelet systems. As we can see from there, the tensor product construction of wavelet systems is extremely simple. This is one of the major reasons the tensor product has been so popular in constructing multi-D wavelets in practice. However the separable wavelet systems have limitations: (i) they have a strong directional bias along lines parallel to the coordinate directions, (ii) they are not very local[1].

Our goal in this paper is to present an alternative method to the tensor product

[1]One way to measure the localness of a wavelet system is to compute the sum of the volumes of the supports of its mother wavelets (cf. [28, 29]).

for constructing decomposable multi-D refinement masks. We call the new method as *coset sum*. We show that, under an appropriate circumstance, the coset sum shares many attractive features of the tensor product. First, it preserves the biorthogonality of univariate refinement masks. Second, it preserves the accuracy number of the univariate refinement mask. Third, it has a corresponding wavelet system which has fast algorithms for computing and inverting the wavelet coefficients. In fact, it turns out that these algorithms are faster, in certain cases, than the known algorithms based on tensor product wavelet systems.

Let us elaborate on the last point in more detail. Suppose that we consider two wavelet systems that are constructed from the same pair of 1-D biorthogonal refinement masks, by using tensor product and coset sum. For the tensor product wavelet system, the associated algorithm has complexity $(\alpha + \beta)nN$ (cf. Section §2.2.2), where α and β are the number of nonzero coefficients of the 1-D lowpass filters for decomposition and reconstruction, respectively, n is the spatial dimension, and N is the size of an initial data to be analyzed. Thus, the constant in the complexity bound (cf. Complexity discussion in Section §2.4.2 for the definition) in this case is $(\alpha + \beta)n$ and it grows linearly with the spatial dimension. On the other hand, as we can see from Section §2.4.2, the complexity constant of the algorithm associated with the coset sum wavelet system we construct in this paper has complexity constant $\frac{3}{2}\alpha + 2\beta$, which is smaller than $(\alpha + \beta)n$ as long as $n \geq 2$. We note that the complexity constant for the coset sum case does not increase even if the spatial dimension increases. For more details, we refer to Section §2.4.2.

The main difference between the coset sum method and the tensor product method is that a "sum" is used in obtaining the coset sum multi-D refinement masks instead of a

"product" used in the tensor product refinement masks. Another difference is that, on the contrary to the tensor product case, the coset sum refinement mask cannot be decomposed into non-univariate refinement masks. Table 2.1 summarizes the comparison between the tensor product and the coset sum.

Some experimental results using 2-D images are included to show the potential usefulness of the coset sum wavelet systems we construct in this paper (cf. Section §2.4.3). They show that our wavelet systems can be potentially useful for effectively approximating a certain class of images with strong directional content. They also reveal some of the limitations of our wavelet systems, which include the lack of rotational symmetry [30]. For details, we refer to Section §2.4.3.

The rest of the paper is organized as follows. In Section §2.2 we briefly overview some relevant concepts on wavelet construction. In Section §2.3 we introduce the coset sum method and discuss its properties. In Section §2.4 we also introduce a particular class of coset sum wavelet systems, together with the associated fast algorithms and some experimental results using our wavelet systems. We summarize our results and present some observations in Section §2.5. Appendix §2.6 contains technical details including all the proofs of the theorems in this paper.

2.2 Preliminaries

In this section we review some relevant concepts.

Table 2.1: Comparison between tensor product and coset sum (spatial dimension $n \geq 2$)

Tensor product $\mathcal{T}_n$ (R, $\tilde{R}$: 1-D refinement masks)	**Coset sum** $\mathcal{C}_n$ (R, $\tilde{R}$: 1-D refinement masks; $\tilde{R}$: *interpolatory*)
$\mathcal{T}_n[R]$ can be decomposed into the *product* of R	$\mathcal{C}_n[R]$ can be decomposed into the *sum* of R
$\mathcal{T}_n[R]$ is interpolatory iff R is interpolatory	$\mathcal{C}_n[R]$ is interpolatory iff R is interpolatory
$\mathcal{T}_n[R]$ and $\mathcal{T}_n[\tilde{R}]$ are biorthogonal iff R and $\tilde{R}$ are biorthogonal	$\mathcal{C}_n[R]$ and $\mathcal{C}_n[\tilde{R}]$ are biorthogonal iff R and $\tilde{R}$ are biorthogonal
$\mathcal{T}_n[R]$ and R have the same accuracy number	$\mathcal{C}_n[\tilde{R}]$ and $\tilde{R}$ have the same accuracy number
$\mathcal{T}_n[R]$ can be decomposed into *non-univariate* refinement masks	$\mathcal{C}_n[R]$ can be decomposed *only* into *univariate* refinement masks
Complexity constant in associated algorithm *increases with* n	Complexity constant in associated algorithm is *independent of* n

2.2.1 Refinement masks and wavelet masks

In this paper we refer to a Laurent trigonometric polynomial as a *mask*, and a mask τ with $\tau(0) = 1$ as a *refinement mask*. Refinement masks can be used to obtain refinable functions (see, for example, [31]), which can in turn be used to construct wavelet systems [1].

Refinement masks τ and τ^{d} are *biorthogonal* if they satisfy the following biorthogonal relation:

$$\sum_{\gamma\in\pi\Gamma} (\overline{\tau}\tau^{\mathrm{d}})(\omega+\gamma) = 1, \quad \forall\omega \in \mathbb{T}^n := [-\pi,\pi]^n, \tag{2.1}$$

where $\Gamma := \{0,1\}^n$ and the overline is used to denote the complex conjugate. In this case, we refer to τ and τ^{d} as *primal and dual refinement masks*, respectively.

A refinement mask τ is *interpolatory* if the condition

$$\sum_{\gamma\in\pi\Gamma} \tau(\omega+\gamma) = 1$$

holds. Thus refinement masks τ and τ^{d} are biorthogonal if and only if $\overline{\tau}\tau^{\mathrm{d}}$ is interpolatory. Interpolatory masks are widely used in subdivision schemes and wavelet constructions (for example, see [32] and references therein).

In this paper we say that a filter $h : \mathbb{Z}^n \to \mathbb{R}$ is *associated with* a mask τ if h and τ are connected via the relation $\tau(\omega) = \frac{1}{2^n}\sum_{k\in\mathbb{Z}^n} h(k)e^{-ik\cdot\omega}$ for $\omega \in \mathbb{T}^n$.

It is straightforward to see that τ is interpolatory if and only if the associated filter h satisfies

$$h(k) = \begin{cases} 1, & \text{if } k = 0, \\ 0, & \text{if } k \in 2\mathbb{Z}^n\backslash 0, \end{cases} \tag{2.2}$$

to which we refer as the interpolatory condition for the filter.

For a refinement mask τ, the number of zeros of τ at $\gamma \in \pi\Gamma'$ with $\Gamma' := \Gamma\backslash 0 = \{0,1\}^n\backslash 0$ is referred to as the *accuracy number* [3]. Throughout the paper we assume that all refinement masks have at least accuracy number one, since almost all of the refinement masks used in practice satisfy this condition.

We recall that the Laurent polynomials $\{t_j, t_j^{\mathrm{d}} : j = 1, \cdots, l\}$ are called the *wavelet masks* associated with a pair of biorthogonal refinement masks $(\tau, \tau^{\mathrm{d}})$ if they satisfy the Mixed Unitary Extension Principle (MUEP) conditions [33]: for every $\omega \in \mathbb{T}^n$,

$$\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega) + \sum_{j=1}^{l} \overline{t_j(\omega+\gamma)} t_j^{\mathrm{d}}(\omega) = \begin{cases} 1, & \text{if } \gamma = 0, \\ 0, & \text{if } \gamma \in \pi\Gamma'. \end{cases} \tag{2.3}$$

We refer to t_j, $j = 1, \cdots, l$, and t_j^{d}, $j = 1, \cdots, l$, as *primal and dual wavelet masks*, respectively. When $l = 2^n - 1$, the masks that satisfy the MUEP conditions can be used to construct biorthogonal wavelet systems. We refer to such $(\tau, (t_j)_{j=1,\cdots,2^n-1})$ and $(\tau^{\mathrm{d}}, (t_j^{\mathrm{d}})_{j=1,\cdots,2^n-1})$ as the *combined biorthogonal masks*. A (MRA-based) *biorthogonal wavelet system* is then obtained from these combined biorthogonal masks, under some simple additional conditions [34, 35].

For a wavelet mask t, the number of zeros of t at $\omega = 0$ is referred to as the *number of (discrete) vanishing moments* [36]. It is well known (see, for example, [36]) that for the combined biorthogonal masks $(\tau, (t_j)_{j=1,\cdots,2^n-1})$ and $(\tau^{\mathrm{d}}, (t_j^{\mathrm{d}})_{j=1,\cdots,2^n-1})$ whose refinement masks have at least m accuracy, every primal wavelet mask t_j and dual wavelet mask t_j^{d}, $j = 1, \cdots, 2^n - 1$, has at least m vanishing moments. The number of vanishing moments is closely related to the approximation performance of the wavelet system [2].

2.2.2 Tensor product wavelet construction

We recall that the n-D tensor product (or separable) refinement mask from n (possibly distinct) univariate refinement masks $R_1, R_2, \cdots, R_n$ can be written as, for $\omega = (\omega_1, \omega_2, \cdots, \omega_n) \in \mathbb{T}^n$,

$$\mathcal{T}_n[R_1, R_2, \cdots, R_n](\omega) := R_1(\omega_1)R_2(\omega_2)\cdots R_n(\omega_n). \tag{2.4}$$

When $R = R_1 = R_2 = \cdots = R_n$, we also use the notation $\mathcal{T}_n[R]$. If we let H and h be the filters associated with the masks R and $\mathcal{T}_n[R]$ respectively, they satisfy, for $k = (k_1, k_2, \cdots, k_n) \in \mathbb{Z}^n$,

$$h(k) = H(k_1)H(k_2)\cdots H(k_n).$$

It is well known that the n-D refinement masks constructed using tensor product preserve many useful properties of univariate refinement masks. For example, if we let R and $\tilde{R}$ be univariate refinement masks, then

(i) $\mathcal{T}_n[R]$ is interpolatory if and only if R is interpolatory,

(ii) $\mathcal{T}_n[R]$ and $\mathcal{T}_n[\tilde{R}]$ are biorthogonal if and only if R and $\tilde{R}$ are biorthogonal,

(iii) $\mathcal{T}_n[R]$ and R have the same accuracy number.

Now we pose the following question. Can we find another method that satisfies all of the above properties? An affirmative answer is provided by the coset sum, which we introduce and study in the next section. Before introducing the coset sum, let us review the usual approach for constructing biorthogonal wavelet systems.

Construction of 1-D biorthogonal wavelet systems is well understood. Given a pair of 1-D biorthogonal refinement masks S_0 and U_0, one sets the wavelet masks as

$$S_1(\omega) := e^{-i\omega}\overline{U_0(\omega+\pi)}, \quad U_1(\omega) := e^{-i\omega}\overline{S_0(\omega+\pi)} \tag{2.5}$$

for $\omega \in \mathbb{T}$. Then the univariate pairs (S_0, S_1) and (U_0, U_1) satisfy the MUEP conditions (cf. (2.3)) [34].

On the other hand, given a pair of multivariate biorthogonal refinement masks, constructing a multivariate biorthogonal wavelet system is not so trivial since one needs to find $2^n - 1$ primal wavelet masks t_j's and $2^n - 1$ dual wavelet masks t_j^{d}'s.

The usual construction of multi-D biorthogonal wavelet systems is done by the tensor product. Given a pair of 1-D biorthogonal refinement masks S_0 and U_0, one sets the n-D refinement masks as

$$\tau := \mathcal{T}_n[S_0], \quad \tau^{\mathrm{d}} := \mathcal{T}_n[U_0]$$

and the n-D wavelet masks as

$$t_\nu = \mathcal{T}_n[S_{\nu_1}, S_{\nu_2}, \cdots, S_{\nu_n}], \quad t_\nu^{\mathrm{d}} = \mathcal{T}_n[U_{\nu_1}, U_{\nu_2}, \cdots, U_{\nu_n}]$$

for all $\nu = (\nu_1, \nu_2, \cdots, \nu_n) \in \Gamma'$. Then the two refinement masks τ and τ^{d} are also biorthogonal, and $(\tau, (t_\nu)_{\nu\in\Gamma'})$ and $(\tau^{\mathrm{d}}, (t_\nu^{\mathrm{d}})_{\nu\in\Gamma'})$ satisfy the MUEP conditions (cf. (2.3)). Here $\Gamma' = \{0,1\}^n \backslash 0$ is used as before, and the univariate masks S_1 and U_1 are the ones defined in (2.5). The biorthogonal wavelet systems obtained from these masks are called *tensor product (or separable) wavelet systems.*

It is well known that tensor product wavelet systems have fast algorithms for computing and inverting wavelet coefficients (see, for example, [37]), to which we refer as

the *fast tensor product wavelet algorithms*. These algorithms have linear complexity $O(N)$, where N is the size of the input data. More precisely, if α is the number of nonzero entries of the filter associated with S_0 and β is the number of nonzero entries of the filter associated with U_0, then the algorithms for computing and inverting the corresponding tensor product wavelet coefficients have complexity $(\alpha + \beta)nN$, where n is the spatial dimension. In particular, the constant in the complexity bound is $(\alpha+\beta)n$ and it increases linearly as the spatial dimension increases.

2.3 Coset sum

2.3.1 Introduction to coset sum

We present an alternative method, called *coset sum*, to the tensor product in wavelet construction. Instead of the "product" in the tensor product, we propose to use a "sum" to construct multivariate refinement masks from univariate refinement masks.

Let R be a univariate refinement mask and let H be the univariate filter associated with R. For $\nu \in \Gamma'$, the map

$$\mathbb{T}^n \to \mathbb{C} : \omega \mapsto \frac{1}{2^{n-1}} R(\omega \cdot \nu),$$

where $\omega \cdot \nu$ is the inner product in $\mathbb{R}^n$, is an n-D Laurent trigonometric polynomial. The normalization factor $\frac{1}{2^{n-1}}$ is used to place $R(\omega \cdot \nu)$ in the n-D space. In terms of filters, the above can be understood as aligning the 1-D filter H along the ν direction:

$$\mathbb{Z}^n \to \mathbb{R} : k \mapsto \begin{cases} H(K), & \text{if } k = K\nu \text{ for some } K \in \mathbb{Z}\ , \\ 0, & \text{otherwise} \end{cases}$$

Since we want to consider all the directions in Γ', a possible candidate for the coset sum definition can be given as

$$\mathbb{T}^n \to \mathbb{C} : \omega \mapsto A + \frac{1}{2^{n-1}} \sum_{\nu \in \Gamma'} R(\omega \cdot \nu).$$

Since we want the coset sum to map a 1-D *refinement* mask to an n-D *refinement* mask, by plugging in $\omega = 0$, we obtain $A = -1 + \frac{1}{2^{n-1}}$ and get to the following definition.

Definition 1 *We define the coset sum $\mathcal{C}_n$ that maps a 1-D refinement mask R to an n-D refinement mask $\mathcal{C}_n[R]$ as follows: for $\omega \in \mathbb{T}^n$*

$$\mathcal{C}_n[R](\omega) := \frac{1}{2^{n-1}} \left(1 - 2^{n-1} + \sum_{\nu \in \Gamma'} R(\omega \cdot \nu) \right),$$

where $\Gamma' = \Gamma \backslash 0 = \{0,1\}^n \backslash 0$. ■

Remark 1. We call the refinement mask obtained by the coset sum method as the *coset sum refinement mask*. The set $\Gamma = \{0,1\}^n$ used in the definition is a complete set of representatives of the distinct cosets (hence the name "coset sum") of the quotient group $\mathbb{Z}^n/2\mathbb{Z}^n$. It is easy to observe that, because n-D masks are 2π-periodic, the set $\{0,1\}^n$ used in this paper prior to the above definition (for example, for biorthogonality condition, interpolatory condition, definition of accuracy number, and MUEP conditions) can be replaced, without changing the meaning of the statements, by any other complete set of representatives of the distinct cosets of the quotient group $\mathbb{Z}^n/2\mathbb{Z}^n$ as long as the set contains 0. As a result, the set $\{0,1\}^n$ used in the above coset sum definition can be replaced by any such an alternative set. The set $\{0,1\}^n$ is chosen for the discussion in this paper (with the exception of Example 3 below and discussions in Section §2.4.3) because it makes the support of the associated filter the smallest. Depending on applications, choosing a different set of representatives

can make more sense. We emphasize that even if all the results in our paper (including Theorem 1, 2, and the fast coset sum wavelet algorithms in later part of the paper) are presented using the set $\Gamma = \{0,1\}^n$, they will stay intact for other choices for Γ. ■

Remark 2. We recall that the sum in the left-hand side of the biorthogonality condition in (2.1) is taken over the set $\pi\Gamma$, which can be considered as a set of coset representatives of $2\pi(\frac{1}{2}\mathbb{Z}^n/\mathbb{Z}^n)$. The set of coset representatives has been previously used in the wavelet literature, mostly in relation with this biorthogonality condition. For example, a new algorithm called a coset by coset (CBC) is proposed in [38] for obtaining dual masks with arbitrary number of accuracy given an interpolatory primal mask, and the coset representatives are used in [39] for an explicit, flexible, and easy implementation of interpolatory subdivision schemes. ■

The coset sum for the first few low dimensions are given as follows:

$$\mathcal{C}_1[R](\omega_1) = R(\omega_1),$$

$$\mathcal{C}_2[R](\omega_1,\omega_2) = \frac{1}{2}\left\{-1 + R(\omega_1) + R(\omega_2) + R(\omega_1+\omega_2)\right\},$$

$$\mathcal{C}_3[R](\omega_1,\omega_2,\omega_3) = \frac{1}{4}\{-3 + R(\omega_1) + R(\omega_2) + R(\omega_1+\omega_2)$$

$$+R(\omega_3) + R(\omega_1+\omega_3) + R(\omega_2+\omega_3) + R(\omega_1+\omega_2+\omega_3)\}.$$

We note that the coset sum formula in the above definition can also be written as

$$-1 + \frac{1}{2^{n-1}}\sum_{\nu\in\Gamma} R(\omega\cdot\nu) \tag{2.6}$$

or

$$\frac{1}{2^{n-1}}\left(\frac{1}{2} + \sum_{\nu\in\Gamma'}\left(R(\omega\cdot\nu) - \frac{1}{2}\right)\right). \tag{2.7}$$

The filter h associated with the coset sum refinement mask $\mathcal{C}_n[R]$ is connected to the univariate filter H via

$$h(k) = \begin{cases} H(K), & \text{if } k = K\nu \text{ for some } K \in \mathbb{Z}\backslash 0, \nu \in \Gamma', \\ 2^n - (2^n - 1)(2 - H(0)), & \text{if } k = 0, \\ 0, & \text{for all other } k \in \mathbb{Z}^n. \end{cases} \tag{2.8}$$

If the univariate filter H associated with R is interpolatory, the n-D filter h associated with $\mathcal{C}_n[R]$ is also interpolatory and it can be expressed as

$$h(k) = \begin{cases} H(K), & \text{if } k = K\nu \text{ for some } K \in \mathbb{Z}, \nu \in \Gamma', \\ 0, & \text{for all other } k \in \mathbb{Z}^n. \end{cases}$$

In particular, the restriction of the n-D filter h to ν direction, for each $\nu \in \Gamma'$, is the 1-D filter H.

Now we give a few very simple examples of constructing multi-D refinement filters from univariate refinement filters.

Example 1: n-D Haar refinement filter: the only filter that can be obtained using either the tensor product or the coset sum. Consider the 2-D Haar refinement filter

$$h(k) = \begin{cases} 1, & \text{if } \ k = (0,0), (1,0), (0,1) \text{ or } (1,1), \\ 0, & \text{otherwise.} \end{cases}$$

Let H be the 1-D Haar refinement filter

$$H(K) = \begin{cases} 1, & \text{if } \ K = 0 \text{ or } K = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Then h can be obtained from H either by

(I) (Tensor Product Case) aligning the filter H along $y = 0$ line (x-axis) and $y = 1$ line (see Figure[2] 2.1(a)), or by

[2]In the figures of filters drawn in this paper, the bold-faced number is used to represent the value of the filter at the origin.

(a) 2-D Haar by tensor product (b) 2-D Haar by coset sum

Figure 2.1: Constructions of 2-D Haar refinement filter (Tensor product and Coset sum) (cf. Example 1)

(II) (Coset Sum Case) aligning the filter H along $y = 0$ line (x-axis), $x = 0$ line (y-axis), and $y = x$ line (see Figure 2.1(b)).

Since the support of the 2-D tensor product refinement filter will always be a rectangle and the support of the 2-D coset sum refinement filter will always be the union of three line segments in different directions, it is easy to see that, up to the integer translation, the 2-D Haar refinement filter is the only 2-D filter that can be obtained using either the tensor product or the coset sum. It is straightforward to show that, for arbitrary spatial dimension n, the n-D Haar refinement filter is the only filter that can be obtained using either the tensor product or the coset sum. ■

Example 2: Refinement filter associated with an n-D piecewise-linear box spline. Let us consider the 2-D refinement filter h associated with a 2-D piecewise-linear box spline [40]:

$$h(k) = \begin{cases} 1, & \text{if } \ k = (0,0), \\ \frac{1}{2}, & \text{if } \ k = \pm(1,0),\ \pm(0,1), \text{ or } \pm(1,1), \\ 0, & \text{otherwise.} \end{cases}$$

Let H be the refinement filter associated with a 1-D piecewise-linear spline:

$$H(K) = \begin{cases} 1, & \text{if } \ K = 0, \\ \frac{1}{2}, & \text{if } \ K = \pm 1, \\ 0, & \text{otherwise.} \end{cases}$$

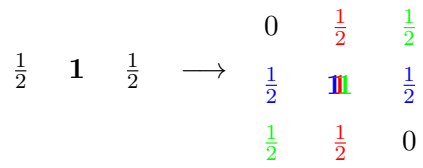


Figure 2.2: Construction of 2-D piecewise-linear box spline refinement filter (Coset sum) (cf. Example 2)

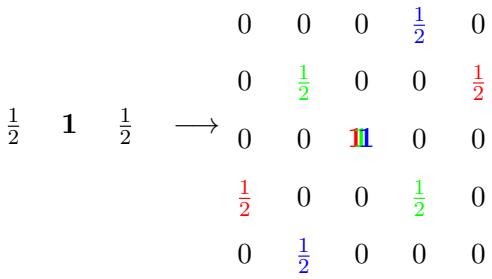


Figure 2.3: 2-D coset sum refinement filter supported on different directed line segments (cf. Example 3)

Then h can be obtained from H by aligning the filter H along $y = 0$ line (x-axis), $x = 0$ line (y-axis), and $y = x$ line (see Figure 2.2). In other words, $h = \mathcal{C}_2[H]$. In fact, it is easy to see that for the n-D refinement filter h associated with an n-D piecewise-linear box spline, we have $h = \mathcal{C}_n[H]$. ■

Example 3: Refinement filter supported on different directed line segments. We consider $n = 2$ and choose the same univariate filter H as in Example 2, but choose the 2-D filter h differently:

$$h(k) = \begin{cases} 1, & \text{if } \ k = (0,0), \\ \frac{1}{2}, & \text{if } \ k = \pm(1,2),\ \pm(2,1), \text{ or } \pm(-1,1), \\ 0, & \text{otherwise.} \end{cases}$$

Then $h = \mathcal{C}_2[H]$ with Γ chosen differently (cf. Remark 1 after Definition 1):

$$\Gamma = \{(0,0), (2,1), (1,2), (-1,1)\}.$$

In particular, h can be obtained from H by aligning the filter H along $y = x/2$ line, $y = 2x$ line, and $y = -x$ line (see Figure 2.3). Note that the filter h is supported on the line segments that are not parallel to the coordinate directions. ■

2.3.2 Properties of coset sum refinement masks

In this subsection, we study the properties of the multi-D refinement masks obtained by the coset sum method.

The following theorem shows that the refinement masks obtained by the coset sum share many important properties with the tensor product refinement masks.

Theorem 1 *Let $\mathcal{C}_n$ be the coset sum, and let R and $\tilde{R}$ be univariate refinement masks.*

(a) $\mathcal{C}_n[R]$ is interpolatory if and only if R is interpolatory.

(b) Suppose that one of R and $\tilde{R}$ is interpolatory. Then $\mathcal{C}_n[R]$ and $\mathcal{C}_n[\tilde{R}]$ are biorthogonal if and only if R and $\tilde{R}$ are biorthogonal.

(c) Suppose that R is interpolatory. Then $\mathcal{C}_n[R]$ and R have the same accuracy number. ■

Proof: See Appendix §2.6.1. ■

Below we add a few remarks on Theorem 1.

Remark on Theorem 1(b). The interpolatory condition in part (b) cannot be omitted. To see this, we consider the univariate refinement mask associated with Daubechies wavelet system of order 2 [41], and let

$$R(\omega) = \tilde{R}(\omega) = \cos^2(\frac{\omega}{2}) \left(\frac{1+\sqrt{3}}{2} + \frac{1-\sqrt{3}}{2} e^{-i\omega} \right), \omega \in \mathbb{T}.$$

Then R (hence $\tilde{R}$) is not interpolatory, and R and $\tilde{R}$ are biorthogonal. However it is easy to see that $\mathcal{C}_2[R]$ and $\mathcal{C}_2[\tilde{R}]$ are not biorthogonal. ■

Remark on Theorem 1(c). For general (not necessarily interpolatory) R, the accuracy number of $\mathcal{C}_n[R]$ is at least $\min\{m_1, m_2\}$ where m_1 is the accuracy number of R and m_2 is the order that $1 - R$ has a zero at the origin. This statement can be proved using similar arguments as in the proof of Theorem 1(c), and we omit the proof. ■

The Deslauriers-Dubuc mask [42] of order $2k$ ($k \in \mathbb{N}$) is defined as

$$\begin{aligned} U_{2k}(\omega) &:= \cos^{2k}(\frac{\omega}{2}) P_k(\sin^2(\frac{\omega}{2})), \\ P_k(x) &:= \sum_{j=0}^{k-1} \frac{(k-1+j)!}{j!(k-1)!} x^j. \end{aligned} \tag{2.9}$$

The mask U_{2k} is interpolatory and has accuracy number $2k$. We now present a family of biorthogonal coset sum refinement masks based on the Deslauriers-Dubuc interpolatory masks.

Example 4: A family of n-D biorthogonal coset sum refinement masks. For each $k \in \mathbb{N}$, we choose U_{2k} in (2.9) as a univariate interpolatory refinement mask. By Theorem 1(a)(c), $\mathcal{C}_n[U_{2k}]$ is an n-D interpolatory refinement mask with accuracy number $2k$. It is straightforward to see that for each $k \in \mathbb{N}$,

$$S_{2k} := U_{2k}(3 - 2U_{2k}) \tag{2.10}$$

is biorthogonal[3] to U_{2k}. By Theorem 1(b), $\mathcal{C}_n[U_{2k}]$ is biorthogonal to $\mathcal{C}_n[S_{2k}]$. Since S_{2k} has at least $2k$ accuracy and $1 - S_{2k}$ has a zero of order at least $2k$ at the origin, by the Remark

[3]Given a refinement filter, a dual refinement filter is not uniquely determined in general. The specific choice of the dual filter of U_{2k} as in (2.10) can be obtained, for example, from Proposition 2.1 in [43]. See also Theorem 2 in [44] for an alternative derivation based on a critical representation of the Laplacian pyramid ([45]).

$$\begin{array}{ccccccc} -\frac{1}{16} & 0 & \frac{9}{16} & \mathbf{1} & \frac{9}{16} & 0 & -\frac{1}{16} \end{array}$$

Filter associated with U_4

$\mathcal{C}_2$ (Coset sum)

$$\begin{array}{ccccccc}
0 & 0 & 0 & -\frac{1}{16} & 0 & 0 & -\frac{1}{16} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{9}{16} & \frac{9}{16} & 0 & 0 \\
-\frac{1}{16} & 0 & \frac{9}{16} & \mathbf{1} & \frac{9}{16} & 0 & -\frac{1}{16} \\
0 & 0 & \frac{9}{16} & \frac{9}{16} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{1}{16} & 0 & 0 & -\frac{1}{16} & 0 & 0 & 0
\end{array}$$

Filter associated with $\mathcal{C}_2[U_4]$

Figure 2.4: Refinement filters associated with the masks U_4 and $\mathcal{C}_2[U_4]$ in Example 4

on Theorem 1(c), $\mathcal{C}_n[S_{2k}]$ has at least $2k$ accuracy. The filters for the case $k = n = 2$ are depicted in Figure 2.4 and 2.5. Using the standard tool in wavelet literature (see, for example, [46] and references therein), one can show that both $\mathcal{C}_2[U_4]$ and $\mathcal{C}_2[S_4]$ generate the refinable functions that are in $L^2(\mathbb{R}^2)$ (cf. Figure 2.6). ■

Similar to the tensor product case, the coset sum can actually take different univariate refinement masks. However, since the cardinality of the set Γ' is $2^n - 1$, we have $2^n - 1$ different directions to consider, instead of n different coordinate directions for the tensor product case. In such a case the n-D coset sum refinement can be written as

$$\mathcal{C}_n[(R_\nu)_{\nu\in\Gamma'}](\omega) := \frac{1}{2^{n-1}}\left(1 - 2^{n-1} + \sum_{\nu\in\Gamma'} R_\nu(\omega\cdot\nu)\right), \tag{2.11}$$

where R_ν, $\nu \in \Gamma'$, are possibly distinct univariate refinement masks for different direction ν.

$$\begin{array}{cccccccccccc} -\frac{2}{512} & 0 & \frac{36}{512} & -\frac{32}{512} & -\frac{126}{512} & \frac{288}{512} & \mathbf{\frac{696}{512}} & \frac{288}{512} & -\frac{126}{512} & -\frac{32}{512} & \frac{36}{512} & 0 & -\frac{2}{512} \end{array}$$

Filter associated with S_4

$\downarrow$ $\mathcal{C}_2$ (Coset sum)

$$\begin{array}{cccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & -\frac{2}{512} & 0 & 0 & 0 & 0 & 0 & -\frac{2}{512} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{36}{512} & 0 & 0 & 0 & \frac{36}{512} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{32}{512} & 0 & 0 & -\frac{32}{512} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{126}{512} & 0 & -\frac{126}{512} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{288}{512} & \frac{288}{512} & 0 & 0 & 0 & 0 & 0 \\
-\frac{2}{512} & 0 & \frac{36}{512} & -\frac{32}{512} & -\frac{126}{512} & \frac{288}{512} & \mathbf{\frac{1064}{512}} & \frac{288}{512} & -\frac{126}{512} & -\frac{32}{512} & \frac{36}{512} & 0 & -\frac{2}{512} \\
0 & 0 & 0 & 0 & 0 & \frac{288}{512} & \frac{288}{512} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{126}{512} & 0 & -\frac{126}{512} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{32}{512} & 0 & 0 & -\frac{32}{512} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{36}{512} & 0 & 0 & 0 & \frac{36}{512} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{2}{512} & 0 & 0 & 0 & 0 & 0 & -\frac{2}{512} & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

Filter associated with $\mathcal{C}_2[S_4]$

Figure 2.5: Refinement filters associated with the masks S_4 and $\mathcal{C}_2[S_4]$ in Example 4

Let $n = n_1 + n_2 + \cdots + n_m, n_j \geq 1$ for $j = 1, 2, \cdots, m$. Then the tensor product refinement mask in (2.4) can be written as the product of possibly non-univariate lower dimensional tensor product refinement masks as follows: for $\omega = (\omega_1, \omega_2, \cdots, \omega_n) \in \mathbb{T}^n$,

$$\begin{aligned} &\mathcal{T}_n[R_1, \cdots, R_n](\omega) \\ &= \mathcal{T}_{n_1}[R_1, \cdots, R_{n_1}](\omega_1, \cdots, \omega_{n_1}) \cdot \end{aligned}$$

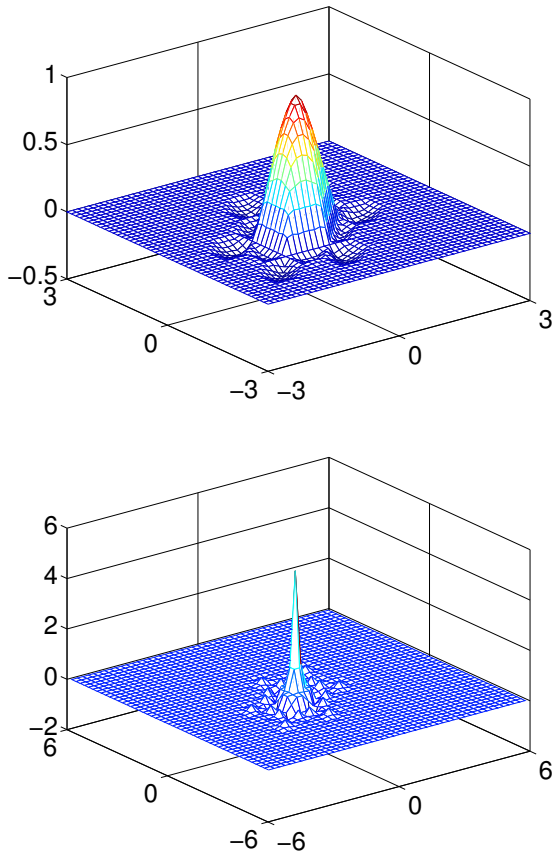


Figure 2.6: The refinable functions associated with the coset sum refinement masks $\mathcal{C}_2[U_4]$ (left) and $\mathcal{C}_2[S_4]$ (right) in Example 4

$$\mathcal{T}_{n_2}[R_{n_1+1},\cdots,R_{n_1+n_2}](\omega_{n_1+1},\cdots,\omega_{n_1+n_2})\cdot$$

$$\cdots\mathcal{T}_{n_m}[R_{n_1+\cdots+n_{m-1}+1},\cdots,R_n](\omega_{n_1+\cdots+n_{m-1}+1},\cdots,\omega_n).$$

On the contrary, the coset sum refinement mask *cannot* be written as the sum of non-univariate lower dimensional coset sum refinement masks.

We can also consider a hybrid of the coset sum and the tensor product : for $n=n_1+n_2+\cdots+n_m, n_j\geq 1, j=1,2,\cdots,m,$

$$\mathcal{C}_{n_1}[R](\omega_1,\cdots,\omega_{n_1})\cdot\mathcal{C}_{n_2}[R](\omega_{n_1+1},\cdots,\omega_{n_1+n_2})\cdot$$

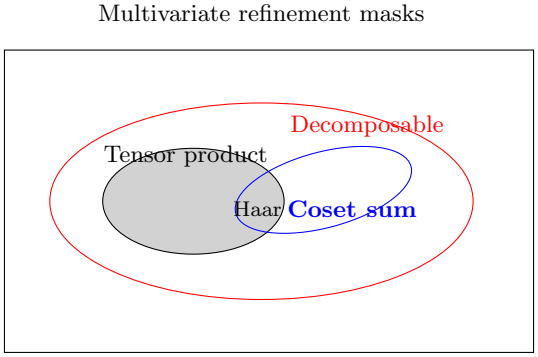


Figure 2.7: The tensor product multivariate refinement masks are not the only decomposable refinement masks. The coset sum provides a systematic way to construct other types of decomposable refinement masks. The other decomposable refinement masks include the ones constructed by the existing approaches (cf. discussion in Section §2.1). The multivariate Haar refinement mask is essentially the only mask that can be obtained by using either the tensor product or the coset sum (cf. Example 1).

$$\cdots \mathcal{C}_{n_m}[R](\omega_{n_1+\cdots+n_{m-1}+1}, \ldots, \omega_n). \tag{2.12}$$

Similar statements to the ones of Theorem 1 can be made for the coset sum refinement mask in a generalized sense as in (2.11) and for the hybrid refinement mask as in (2.12). We omit the statements and the proofs as they are similar to the ones of Theorem 1.

The diagram in Figure 2.7 illustrates the relation among the tensor product, the coset sum, and the decomposable multi-D refinement masks. We note that the type of decomposable refinement masks that can be obtained by coset sum is different from the one by the aforementioned existing methods [21–27] since coset sum works for 2^n-channel sampling lattices (cf. Section §2.4.2).

2.4 Application: coset sum wavelet systems

In this section we introduce a special class of wavelet systems that can be derived from coset sum refinement masks in a very simple manner, and present their properties,

including fast algorithms, together with some experimental results.

2.4.1 Coset sum wavelet systems

Since the coset sum provides a way to construct a pair of multivariate biorthogonal refinement masks from univariate ones, it can be combined with any procedure for finding wavelet masks to construct a multivariate biorthogonal wavelet system. It is well known (for example, see [47] and Example 5 below) that for a given pair of n-D biorthogonal refinement masks, different biorthogonal wavelet systems can be obtained by choosing wavelet masks differently. The specific choice we make in this paper is guided by the simplicity of the form of the primal wavelet masks (cf. (3.4) and the discussion below). Use of other criteria may result in a totally different type of "coset sum" wavelet systems, hence discussing about properties of coset sum wavelet systems makes sense only after the wavelet masks are specifically chosen. Below we present our approach for determining the wavelet masks.

Suppose that S and U are 1-D biorthogonal refinement masks, and that U is interpolatory. Theorem 1(b) implies that the n-D coset sum refinement masks $\mathcal{C}_n[S]$ and $\mathcal{C}_n[U]$ are biorthogonal. Moreover, from (2.7) and the assumption that U is interpolatory, we see that the restriction of the n-D mask $\mathcal{C}_n[U]$ to ν direction, $\nu \in \Gamma' = \{0,1\}^n \backslash 0$, is given by $U(\omega \cdot \nu)$ for $\omega \in \mathbb{T}^n$ (up to constants), which is essentially a 1-D mask. Hence, as in the 1-D wavelet construction (cf. (2.5)), one can attempt to define the multivariate wavelet masks t_ν, $\nu \in \Gamma'$, (note that we have $2^n - 1$ wavelet masks) of the form

$$t_\nu(\omega) = e^{-i\omega\cdot\nu}\overline{U(\omega \cdot \nu + \pi)}, \quad \omega \in \mathbb{T}^n. \tag{2.13}$$

The next theorem shows that the above approach leads to the construction of n-D biorthog-

onal wavelet systems.

Theorem 2 *Suppose that S and U are 1-D biorthogonal refinement masks, and that U is interpolatory. Define n-D biorthogonal refinement masks as*

$$\tau := \mathcal{C}_n[S], \quad \tau^{\mathrm{d}} := \mathcal{C}_n[U],$$

and n-D primal wavelet masks t_ν, $\nu \in \Gamma'$, as in (3.4). Then there exist dual wavelet masks t^{d}_ν, $\nu \in \Gamma'$, such that $(\tau, (t_\nu)_{\nu\in\Gamma'})$ and $(\tau^{\mathrm{d}}, (t^{\mathrm{d}}_\nu)_{\nu\in\Gamma'})$ are n-D combined biorthogonal masks. ■

Proof: See Appendix §2.6.2. ■

Remark 1. We refer to the biorthogonal wavelet system constructed from the n-D combined biorthogonal masks in Theorem 2 as the *canonical coset sum wavelet system*. As we discussed previously, there may be many other coset sum wavelet systems associated with the same coset sum refinement masks. Throughout this paper, the word "canonical" is suppressed when no confusion arises. ■

Remark 2. The exact form of the dual wavelet masks t^{d}_ν, $\nu \in \Gamma'$, of the canonical coset sum wavelet system in Theorem 2 is not important for understanding our results in this paper, but knowing it may be useful in some other contexts. By carefully inspecting the proof of Theorem 2, we see that the dual wavelet masks t^{d}_ν, $\nu \in \Gamma'$, have the form

$$t^{\mathrm{d}}_\nu(\omega) = 2^{-n+1} e^{-i\omega\cdot\nu}(1 - 2\tau^{\mathrm{d}}(\omega)\overline{S^o(\omega\cdot\nu)}),\ \omega \in \mathbb{T}^n, \tag{2.14}$$

where $S^o := (S - S(\cdot + \pi))/2$ is the odd part of S. ■

Remark 3. We recall that there is a nonseparable multi-D wavelet construction method based on the traditional lifting scheme ([48]) proposed by J. Kovačević and W. Sweldens

[49]. A key ingredient of their construction is a class of n-D filters called Neville filters, which are used to build the predict and update filters. Given suitable n-D Neville filters, their method can construct the associated n-D biorthogonal wavelet systems. It turns out that if the Neville filters are extracted from the n-D biorthogonal coset sum refinement masks, the above canonical coset sum wavelet systems can also be obtained by their method, and our fast algorithms associated with these wavelet systems (cf. Section §2.4.2) can be viewed as realization of a special case of their fast transform. However it should be noted that the Neville filters extracted from the n-D biorthogonal coset sum refinement masks cannot be obtained from [49]. Also note that other coset sum wavelet systems besides the canonical ones cannot be constructed by their method regardless of the choice of the Neville filters.

■

Canonical coset sum wavelet systems have many potentially useful properties. The most distinctive property is that they can be associated with fast algorithms, which is explained in detail in the next subsection. Another (related) property is that they can be much more local than tensor product wavelet systems. The easiest way to see this property is probably through the following example.

Example 5: n-D coset sum Haar wavelet systems. The simplest choice for the univariate refinement mask is the 1-D Haar refinement mask $R(\omega) = \frac{1}{2} + \frac{1}{2}e^{-i\omega}$, $\omega \in \mathbb{T}$, which is biorthogonal to itself and interpolatory. Let $\tau = \tau^{\mathrm{d}} = \mathcal{C}_n[R]$ be the n-D Haar refinement mask, which can be obtained either by coset sum or tensor product (cf. Example 1). Then from Theorem 2 and the remarks after it, we obtain the canonical coset sum wavelet system whose n-D biorthogonal refinement masks are τ and τ^{d}, and whose primal and dual wavelet

masks are, for $\nu \in \Gamma'$ and $\omega \in \mathbb{T}^n$,

$$t_\nu(\omega) = \frac{e^{-i\omega\cdot\nu} - 1}{2}, \quad t^{\mathrm{d}}_\nu(\omega) = \frac{e^{-i\omega\cdot\nu} - \tau^{\mathrm{d}}(\omega)}{2^{n-1}}.$$

We refer to this wavelet system as the n-D (canonical) coset sum Haar wavelet system. Each of the wavelet masks of this wavelet system has one vanishing moment. We note that this wavelet system is the same as the piecewise-constant biorthogonal wavelet system introduced in [28] (up to constants), which is shown to be far more local than the tensor product Haar wavelet system, in high spatial dimensions. ■

Remark. We recall that orthogonality is a special case of biorthogonality. We note that the n-D Haar refinement mask (cf. Example 1 and 5) is orthogonal, whereas the n-D canonical coset sum Haar wavelet system (cf. Example 5) is not orthogonal. In fact, it is not possible to construct n-D canonical coset sum wavelet system that is orthogonal. This can be seen from the facts that the 1-D refinement mask we start with for such a wavelet system has to be interpolatory and orthogonal, and that there is no 1-D interpolatory orthogonal refinement mask (in the dyadic dilation) other than the Haar one (see, for example, [43]), whose associated n-D canonical coset sum wavelet system is not orthogonal as we just established. ■

A drawback of the n-D coset sum Haar wavelet system in the previous example is that the wavelet masks have only one vanishing moment. In order to construct n-D biorthogonal wavelet systems with larger number of vanishing moments, one needs to have n-D biorthogonal refinement masks with larger number of accuracy (cf. Section §2.2.1). In general, constructing n-D biorthogonal refinement masks with large number of accuracy can be cumbersome, especially when n is large, since it involves solving a large number of linear

$$\begin{array}{ccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{16} & 0 & -\frac{\mathbf{9}}{\mathbf{16}} & 1 & -\frac{9}{16} & 0 & \frac{1}{16} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

(a) Wavelet filter for mask $t_{1,0}(\omega_1, \omega_2)$

$$\begin{array}{ccccccccc}
0 & 0 & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{9}{16} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{\mathbf{9}}{\mathbf{16}} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

(b) Wavelet filter for mask $t_{0,1}(\omega_1, \omega_2)$

$$\begin{array}{ccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{9}{16} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{\mathbf{9}}{\mathbf{16}} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

(c) Wavelet filter for mask $t_{1,1}(\omega_1, \omega_2)$

Figure 2.8: Primal coset sum wavelet filters of Example 6 for 2-D with 4 vanishing moments ($n = k = 2$)

equations. Since coset sum can preserve the biorthogonality and the accuracy number simultaneously, it allows one to bypass solving these linear systems to get biorthogonal refinement masks with large number of accuracy. Thus it is often easier to construct n-D wavelet systems based on the coset sum than other n-D wavelet systems, for large number of vanishing moments. The next is an example of such coset sum wavelet systems.

Example 6: A family of n-D coset sum wavelet systems with larger number of

vanishing moments. We choose the univariate refinement masks U_{2k} (interpolatory) and S_{2k} as in (2.9) and (2.10), respectively, and apply Theorem 2. Then with the primal wavelet masks given as

$$t_\nu(\omega) = e^{-i\omega\cdot\nu}\sin^{2k}(\frac{\omega\cdot\nu}{2})P_k(\cos^2(\frac{\omega\cdot\nu}{2})), \quad \nu\in\Gamma',$$

with $\Gamma' = \{0,1\}^n\backslash 0$, there exist dual wavelet masks $t^{\rm d}_\nu$, $\nu\in\Gamma'$, such that $(\mathcal{C}_n[S_{2k}], (t_\nu)_{\nu\in\Gamma'})$ and $(\mathcal{C}_n[U_{2k}], (t^{\rm d}_\nu)_{\nu\in\Gamma'})$ are n-D combined biorthogonal masks. It is easy to check that each of the wavelet masks of this wavelet system has $2k$ vanishing moments. All the primal wavelet filters are supported on the union of 2^n-1 line segments along ν direction for $\nu\in\Gamma'$. For example, if $n=2$, then $\Gamma' = \{(1,0),(0,1),(1,1)\}$ and the primal wavelet masks for the case $k=2$ are given as

$$\begin{aligned}
t_{(1,0)}(\omega_1,\omega_2) &= e^{-i\omega_1}\sin^4(\frac{\omega_1}{2})\left(1+2\cos^2(\frac{\omega_1}{2})\right),\\
t_{(0,1)}(\omega_1,\omega_2) &= e^{-i\omega_2}\sin^4(\frac{\omega_2}{2})\left(1+2\cos^2(\frac{\omega_2}{2})\right),\\
t_{(1,1)}(\omega_1,\omega_2) &= e^{-i(\omega_1+\omega_2)}\sin^4(\frac{\omega_1+\omega_2}{2})\\
&\quad\cdot\left(1+2\cos^2(\frac{\omega_1+\omega_2}{2})\right).
\end{aligned}$$

The associated wavelet filters are depicted in Figure 2.8. The magnitude of the primal masks $\mathcal{C}_2[S_4], t_{(1,0)}, t_{(0,1)}$, and $t_{(1,1)}$ (i.e. the magnitude of the frequency responses of the filters associated with the primal masks) are depicted in Figure 2.9. The magnitude of the corresponding tensor product primal masks are given in Figure 2.10 for comparison.

Since $\mathcal{C}_2[U_4]$ and $\mathcal{C}_2[S_4]$ generate refinable functions in $L^2(\mathbb{R}^2)$ (cf. Example 4 and Figure 2.6), and since we have FIR filters, the 2-D coset sum wavelet system generated from the combined biorthogonal masks $(\mathcal{C}_2[S_4], t_{(1,0)}, t_{(0,1)}, t_{(1,1)})$ and $(\mathcal{C}_2[U_4], t^{\rm d}_{(1,0)}, t^{\rm d}_{(0,1)}, t^{\rm d}_{(1,1)})$ is also in $L^2(\mathbb{R}^2)$. ■

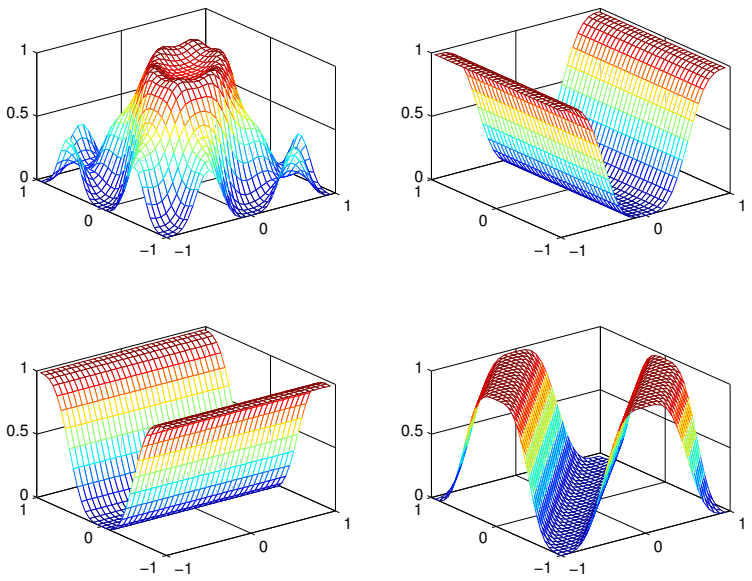


Figure 2.9: The magnitude of the primal coset sum masks $\mathcal{C}_2[S_4], t_{(1,0)}, t_{(0,1)}$, and $t_{(1,1)}$ in Example 6

It should be noted that the above properties (the space localization property discussed in Example 5, and the frequency responses, the vanishing moments, and the smoothness–in the sense of whether or not a wavelet system belongs to L^2–discussed in Example 6) of canonical coset sum wavelet systems may not hold true for other coset sum wavelet systems.

2.4.2 Fast coset sum wavelet algorithms

Next we show that the canonical coset sum wavelet system can be associated with the fast algorithm with linear complexity whose complexity constant does not grow with the spatial dimension. When presenting and analyzing our algorithm below, we use mostly filters instead of masks that we have been used so far, as this approach will be more useful

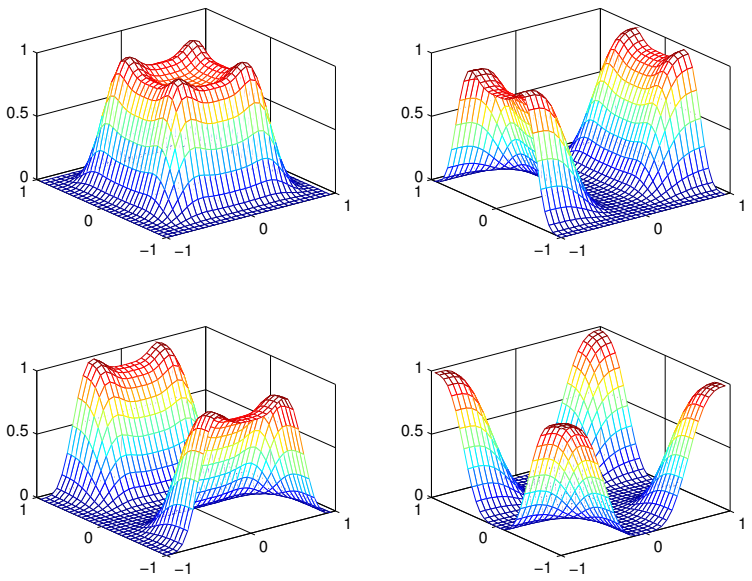


Figure 2.10: The magnitude of the primal tensor product masks that are comparable to the masks in Figure 2.9

in practice.

Fast Coset Sum Wavelet Algorithms. Let S and U be biorthogonal univariate refinement masks, where U is interpolatory. Let G and H be the filters associated with the refinement masks S and U, respectively. In particular, H is interpolatory (cf. (2.2)).

input $y_J : \mathbb{Z}^n \to \mathbb{R}$

(1) Decomposition Algorithm:

$a_G = -2^n + 2 + (2^n - 1)G(0)$

`for` $j = J, J-1, \cdots, 1$

`for` $k \in \mathbb{Z}^n$

$$y_{j-1}(k) = \frac{1}{2^n}(a_G y_j(2k) + \sum_{\nu \in \Gamma'} \sum_{L \in \mathbb{Z} \setminus 0} G(L) y_j(2k + L\nu)) \qquad \texttt{(i)}$$

```
 end
 for ν ∈ Γ′ and k ∈ ℤ^n
```

$$w_{\nu,j-1}(k) = \frac{1}{2}(y_j(2k+\nu) - \sum_{m\equiv 1} H(m) y_j(2k+(1-m)\nu)) \qquad \text{(ii)}$$

```
 end
end
```

(2) Reconstruction Algorithm:

```
for j = 1, ⋯, J − 1, J
 for k ∈ ℤ^n
```

$$y_j(2k) = y_{j-1}(k) - \frac{1}{2^{n-1}} \sum_{\nu\in\Gamma'} \sum_{L\in\mathbb{Z}} G(2L+1) w_{\nu,j-1}(k+L\nu) \qquad \text{(iii)}$$

```
 end
 for ν ∈ Γ′ and k ∈ ℤ^n
```

$$y_j(2k+\nu) = 2w_{\nu,j-1}(k) + \sum_{m\equiv 1} H(m) y_j(2k+(1-m)\nu) \qquad \text{(iv)}$$

```
 end
end
```

Given coarse coefficients y_j at level j, Decomposition Algorithm first computes the lower level coarse coefficients y_{j-1}, and then the wavelet coefficients $w_{\nu,j-1}$, $\nu \in \Gamma' = \{0,1\}^n \backslash 0$. The coefficients y_{j-1} and $w_{\nu,j-1}$ are obtained by filtering (using the n-D filter g associated with the coset sum refinement mask $\mathcal{C}_n[S]$ for y_{j-1} and the n-D filter h_ν associated with the primal coset sum wavelet mask t_ν for $w_{\nu,j-1}$) followed by downsampling, as is typically done in wavelet decomposition process (see, for example, [41]). Since the n-D mask $\mathcal{C}_n[S]$ can be written in terms of 1-D mask S (cf. Definition 1), the associated n-D filter g can be written in terms of 1-D filter G (cf. (3.7)). Similarly, from the fact that the

n-D mask t_ν can be written in terms of 1-D mask U (cf. (3.4)), the associated n-D filter h_ν can be written in terms of 1-D filter H. Taking into account these observations, we get the above simple expressions for y_{j-1} and $w_{\nu,j-1}$ in Step `(i)` and `(ii)`, respectively. In Step `(ii)` and `(iv)`, $m \equiv 1$ is used to mean that m is congruent to 1 in modulo 2, i.e., m is an odd integer.

Reconstruction Algorithm recovers y_j from y_{j-1}, and $w_{\nu,j-1}$, $\nu \in \Gamma'$. It first recovers y_j at even points (cf. Step `(iii)`) and then at all other points (cf. Step `(iv)`). Step `(iii)` is a key step in making our algorithm fast (cf. Complexity discussion below). It is easy to show that the identity in Step `(iii)` holds true for our canonical coset sum wavelet system (see Appendix §2.6.3 for proof), but it need not be true for other coset sum wavelet systems. Step `(iv)` is simply a reverse process of Step `(ii)` and is possible since the only y_j values we need at this step are the values at even points, and these are already computed in Step `(iii)`.

For a given pair of 1-D masks S and U that generates the canonical coset sum wavelet system, the filters H and G in the algorithm can be computed easily. For example, for the n-D coset sum Haar wavelet system in Example 5, both H and G are the 1-D Haar refinement filter (cf. Example 1). For the coset sum wavelet system in Example 6 that is

generated from S_4 (cf. (2.10)) and U_4 (cf. (2.9)), the filters are given as

$$H(K) = \begin{cases} 1, & K = 0, \\ \frac{9}{16}, & K = \pm 1, \\ -\frac{1}{16}, & K = \pm 3, \\ 0, & \text{otherwise}, \end{cases} \qquad G(K) = \begin{cases} \frac{696}{512}, & K = 0, \\ \frac{288}{512}, & K = \pm 1, \\ -\frac{126}{512}, & K = \pm 2, \\ -\frac{32}{512}, & K = \pm 3, \\ \frac{36}{512}, & K = \pm 4, \\ -\frac{2}{512}, & K = \pm 6, \\ 0, & \text{otherwise}. \end{cases}$$

We note that the above algorithms for the canonical coset sum wavelet systems are not redundant: the number of coefficients after the decomposition algorithm is approximately the same as the number of input samples, assuming that the filter length of each filter involved in the algorithm is negligible compared to the number of input samples.

Complexity. We measure complexity by counting the number of operations needed in order to fully derive y_{j-1}, and $w_{\nu,j-1}$, $\nu \in \Gamma'$, from y_j, and add the number of operations needed for the reconstruction. Here, we count only multiplicative operations such as multiplication and division, as counting additive operations gives a similar result.

As in the fast tensor product wavelet algorithms discussed in Section §2.2.2, the complexity here is linear, i.e. $\sim CN$, with N the number of nonzero entries in y_J, and C some constant independent of y_J. We refer to this constant as the *constant in the complexity bound* or simply as the *complexity constant* throughout this paper.

We now estimate the complexity constant for fast coset sum wavelet algorithms by computing the mean number of operations per single entry in y_J. Suppose α and β are the numbers of nonzero entries of the filters G and H, respectively. Then, the number of

operations that are needed to process the portion of y_J that lies on the vertices of a unit cube is the sum of

- $2n-1$ (for computing a_G),
- $(2^n-1)(\alpha-1)+n+1$ (for Step `(i)`),
- $2(2^n-1)\beta$ (for Step `(ii)` and `(iv)`), and
- $(2^n-1)\frac{\alpha+1}{2}+n-1$ (for Step `(iii)`).

After computing the sum, we divide it by 2^n, which is the number of vertices in the unit cube, in order to obtain the cost per entry of performing one complete cycle of decomposition/reconstruction. As a result, we get

$$\frac{3}{2}\alpha+2\beta$$

as an upper bound for the cost per entry. Therefore, the algorithm has complexity $(\frac{3}{2}\alpha+2\beta)N$, and the constant in the complexity bound in this case is $\frac{3}{2}\alpha+2\beta$, which does not increase as the spatial dimension n increases. A similar argument is used in [29] to compute the complexity constant for the algorithm introduced there.

Contrary to the complexity constant of the fast coset sum wavelet algorithm that we just computed, in the tensor product case the constant grows with the dimension (cf. Section §2.2.2). There are a couple of components that make the coset sum wavelet algorithm this fast. First, as we discussed in Section §2.4.1 (cf. (3.4)), the wavelet masks of the coset sum wavelet system are essentially univariate. Second, as we can see from the above algorithms (cf. Step `(iii)`), the reconstruction step can be done by completely bypassing the

dual wavelet filters. This is reminiscent of the Laplacian pyramid [45] (cf. Appendix §2.6.2) and its variant [44], which have trivial reconstruction steps that are simply reverse processes of decomposition steps. As a consequence, our algorithm inherits an asymmetry in the roles of the lowpass filters from the Laplacian pyramid. Hence the 1-D lowpass filters G and H in our fast coset sum wavelet algorithm play different roles. ■

Remark 1. It is well known that any (MRA-based) biorthogonal wavelet system (associated with FIR filters) has decomposition and reconstruction algorithms with linear complexity (see, for example, [1, 3, 34, 37]). In fact, as we alluded to earlier, our fast coset sum decomposition algorithm is nothing but this generic decomposition algorithm for the given canonical coset sum wavelet system. However, our fast coset sum reconstruction algorithm is fundamentally different from this generic reconstruction algorithm: the dual coset sum wavelet filters (cf. (3.9)) that are not used for our reconstruction algorithm are used for the generic one. As a result, our canonical coset sum wavelet system in Theorem 2 has two different algorithms (the fast coset sum wavelet algorithm and the generic one) and the generic algorithm is always slower than the fast coset sum wavelet algorithm. ■

Remark 2. For any biorthogonal wavelet system, multiplying the primal part with some constant factors and dividing the dual part with the same factors will still make a biorthogonal wavelet system. As the functions in these two systems differ only by constants, it is clear that the two systems are essentially the same and most of their properties–including the support and the smoothness–are kept the same.

For the above fast coset sum wavelet algorithms, this means that the decomposition

step can be rewritten with explicit *normalization factors* $c, d > 0$ as

$$y_{j-1}^{new}(k) = cy_{j-1}(k), \quad w_{\nu,j-1}^{new}(k) = dw_{\nu,j-1}(k)$$

where $y_{j-1}(k)$, $w_{\nu,j-1}(k)$ are defined as in Step `(i)-(ii)`, and that the reconstruction step can be modified accordingly: the expressions in the right-hand side of Step `(iii)-(iv)` can be rewritten in terms of

$$\frac{1}{c}y_{j-1}^{new}(k), \quad \frac{1}{d}w_{\nu,j-1}^{new}(k)$$

in place of $y_{j-1}(k)$, $w_{\nu,j-1}(k)$ that are currently used. In this sense, our original fast coset sum wavelet algorithms can be considered as a special case when $c = d = 1$. These normalizations are used throughout this paper except in Section §2.4.3 (see the discussion below and the footnote in the subsection).

When normalization factors are used for the wavelet algorithms, most properties of the algorithms are not affected. For example, fast coset sum wavelet algorithms with normalization factors will still have the linear complexity with the complexity constant that is independent of n. However the use of different normalization factors may result in different performance in practice [50]. For example, when the algorithms are used for nonlinear approximation with multiple levels (cf. Section §2.4.3), the coefficients are multiplied by constant factors and these factors propagate recursively to other coefficients in lower levels and, as a result, the use of normalization factors may change the relative size of the coefficients. ■

Below we compare the fast tensor product wavelet algorithms with the fast coset sum wavelet algorithms, both based on the Deslauriers-Dubuc mask and its dual mask in Section §2.3.2.

Example 7: Fast tensor product wavelet algorithms vs. fast coset sum wavelet algorithms. In this example, we compare the algorithms for two different families of n-D wavelet systems constructed from the same univariate refinement masks by using two different methods: (I) the tensor product and (II) the coset sum. We consider the same univariate refinement masks as in Example 4 and 6, i.e. U_{2k} (interpolatory) and S_{2k} as in (2.9) and (2.10), respectively. It is easy to see that the number of nonzero entries of the filter associated with S_{2k} is $\alpha = 8k - 3$, and the number of nonzero entries of the filter associated with U_{2k} is $\beta = 2k + 1$.

Then complexity constant for each algorithm is given as follows:

(I) (Tensor Product Case) From Section §2.2.2, the complexity constant for the fast tensor product algorithm is $(\alpha + \beta)n = (10k - 2)n$, which grows linearly with the dimension.

(II) (Coset Sum Case) From the above Complexity discussion, the complexity constant for the fast coset sum wavelet algorithm is $\frac{3}{2}\alpha + 2\beta = \frac{3}{2}(8k-3) + 2(2k+1) = 16k - \frac{5}{2}$, which *does not* grow with the dimension.

Therefore, remarkably, if we fix k (hence the number of vanishing moments of the wavelet system) and increase the dimension n, then the *complexity constant stays the same for the coset sum case, whereas it increases for the tensor product case.* ■

2.4.3 Experiments

In this subsection we present some experimental results of the canonical coset sum wavelet system, in comparison with the tensor product wavelet system. We have imple-

mented the fast coset sum wavelet algorithms in Matlab. The program takes a pair of 1-D biorthogonal refinement filters as input and works for 2-D images. We compare our Matlab program with the standard Matlab implementation of 2-D fast tensor product wavelet algorithms: *wavedec2* (for decomposition) and *waverec2* (for reconstruction) in Wavelet Toolbox [51]. [4]For the experiments in this subsection, we use two different 2-D wavelet systems obtained from the same 1-D filters, U_4 (shown in Figure 2.4) and S_4 (shown in Figure 2.5), but using two different methods, coset sum and tensor product. For the coset sum wavelet system, we initially choose $\Gamma' = \{(1,0),(0,1),(1,1)\}$ as the nonzero coset representatives. The two wavelet systems constructed this way are discussed in Example 6, and the complexity constants of their algorithms are compared in Example 7.

We first compare the running time of the fast coset sum wavelet algorithm with the fast tensor product wavelet algorithm. We apply the two wavelet systems constructed as above to test images, "part of lena"[5] in Figure 2.11(a), and "wood45"[6] in Figure 2.12(a), both of which have directional content along the diagonal direction. Here, the diagonal direction, or 45° from the positive x-axis, is chosen because it can highlight the benefit of our coset sum wavelet system over the tensor product wavelet system: it is one of the directions that may be captured well by our coset sum system since $\tan 45^\circ = \frac{1}{1}$ and $(1,1) \in \Gamma'$, while it is one of the directions that may not be captured well by the tensor product system since

[4]When comparing the implementation of two different wavelet systems, it is important to use the same normalization factors as they may affect the performance (cf. Remark 2 after Complexity discussion). Normalization factors $c = d = 2$ are used for implementing our coset sum wavelet system since these are the normalization factors used for the tensor product Matlab implementation when seen in terms of a 2-D generalization of the related 1-D concepts (i.e. the DC and Nyquist gains) [52].

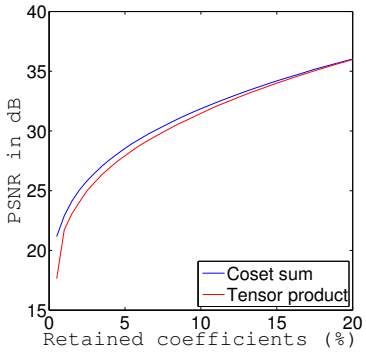
[5]This image is obtained from the image "lena" (512×512) in the image repository http://links.uwaterloo.ca/Repository.html by taking its central part (of size 256×256).

[6]This image is obtained from the image "wood.000" (512×512) in the SIPI Image Database http://sipi.usc.edu/database/database.php?volume=rotate by rotating 45° clockwise, and taking its central part (of size 256×256).

it is not a coordinate direction. Both are of size 256×256, and we perform 5-level-down decomposition and reconstruction. The running time for "part of lena" is about 0.0279 seconds (s) on average for tensor product algorithm and about 0.0161 s on average for coset sum algorithm, on a Mac 4G 1333MHz laptop. The running time for "wood45" is about 0.0283 s on average for tensor product and about 0.0162 s on average for coset sum. We also tried several other images, both with and without directional content, for various levels of decomposition and reconstruction, and obtained essentially the same results: the coset sum algorithms were faster than the tensor product ones. These experiments confirm our theoretical finding in the previous subsection (cf. Example 7).

We now compare the approximation power of these two wavelet systems. For this, we first decompose a fixed image using the two wavelet systems, then recover the image from the M-largest decomposed coefficients (in magnitude), and finally compare the Peak-Signal-to-Noise-Ratio (PSNR) of the two reconstructed images. The reconstructed image with higher PSNR indicates better approximation to the original image. For the image "part of lena", the reconstructed image using coset sum system shows sharper edges along the diagonal direction and better visual quality than those of tensor product system, and has a slightly higher PSNR (Figure 2.11(a)(c)(d)). In this experiment, we found that as long as the percentage of retained coefficients is not too large, the coset sum system has slightly higher PSNR (see Figure 2.11(b) for the range 0.5%-20%). For higher percentage, the coset sum system showed either comparable or slightly worse performance. Another example using the texture image "wood45" is also presented (Figure 2.12). The reconstructed image using coset sum system shows even better performance in this example in terms of PSNR, which is

(a) Original image "part of lena"



(b) PSNR of reconstructed image

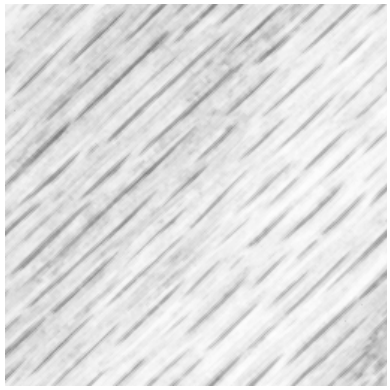
(c) Reconstructed by tensor product

(d) Reconstructed by coset sum

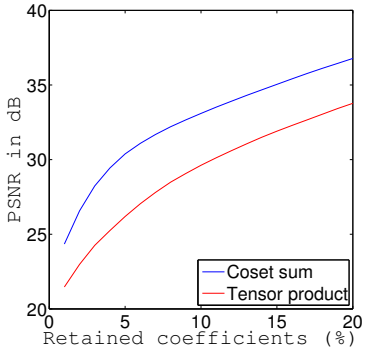
Figure 2.11: Comparison of approximation power of tensor product and coset sum for (a) original image "part of lena": 5-level-down decomposition and reconstruction using 3% largest coefficients. (c) The reconstructed image by tensor product, PSNR = 25.7 dB. (d) The reconstructed image by coset sum with $\Gamma' = \{(1,0),(0,1),(1,1)\}$, showing sharper edges and better visual quality, with improved PSNR = 26.5 dB. (b) PSNR of reconstructed images over different percentage of retained coefficients (0.5%-20%). This experiment shows that the reconstructed images by coset sum have higher PSNR (solid blue), hence better approximation quality than those by tensor product (dotted red) over the range 0.5%-20% for image "part of lena".

probably due to its stronger directional content. Contrary to the previous experiment with "part of lena", in this experiment, the coset sum system showed consistently higher PSNR for all the percentages. From this experiment, we see that coset sum wavelet system shows promising results when applied to images with strong directional content that matches with

the directions of the coset sum primal wavelet filters.

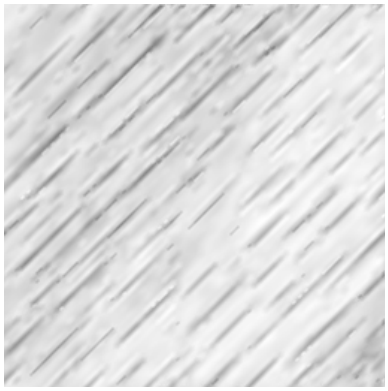


(a) Original image "wood45"



(b) PSNR of reconstructed image

(c) Reconstructed by tensor product



(d) Reconstructed by coset sum

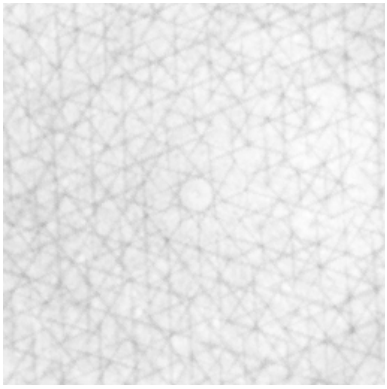
Figure 2.12: Comparison of approximation power of tensor product and coset sum for (a) original image "wood45": 5-level-down decomposition and reconstruction using 3% largest coefficients. (c) The reconstructed image by tensor product, with blurry recovered content and PSNR = 24.3 dB. (d) The reconstructed image by coset sum with $\Gamma' = \{(1,0),(0,1),(1,1)\}$, showing better approximation to the original image and better improved PSNR = 28.2 dB. (b) PSNR of reconstructed images over different percentage of retained coefficients (0.5%-20%). The improvement of PSNR in this example is larger than that in "part of lena" example due to the stronger directional content in image "wood45".

We recall that the directional preference of the coset sum primal wavelet filters can be specified by the associated coset representatives in Γ'. If a dominant direction of the given image does not match with the preferred directions of the coset sum, the coset sum wavelet

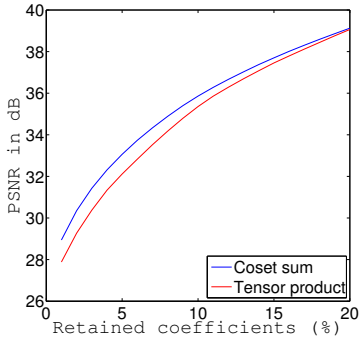
system may no longer perform well. In such a case, a different set Γ' may be used to match the image's direction (cf. Remark 1 after Definition 1 in Section §2.3.1). For example, if the dominant direction is $-60°$ from the positive x-axis, then since $\tan(-60°) = -\sqrt{3} \approx \frac{2}{(-1)}$, the coset representative $(-1, 2)$ can be used in place of $(1, 0)$ in the default nonzero coset representatives $\Gamma' = \{(1, 0), (0, 1), (1, 1)\}$.

For many images, it may not be possible to match the directions of the image with the directions of the coset sum. In order to see how the coset sum system would perform for these images in comparison with the tensor product system, we apply the two systems to the test image, "wood"[7] in Figure 2.13(a). It has 5 different directional content ($25°, 60°$, $95°$, $130°$ and $165°$ from the positive x-axis), and it is impossible for us to choose the nonzero coset representatives that match all the directions presented in the image. The reconstructed image using coset sum with the default coset representatives shows sharper edges along the directions near $45°$, such as $60°$ and $25°$, and better visual quality than those of tensor product system, and has a slightly higher PSNR (see Figure 2.13(a)(c)(d)). For the directions that are significantly different from the preferred directions of the coset sum, such as $130°$, it does not show sharp edges anymore, but the reproduced image using tensor product does not show sharp edges either. We found that the overall PSNR result of this image is similar to that of "part of lena": the reconstructed image by coset sum has slightly higher PSNR as long as the percentage of retained coefficients is not too large (see Figure 2.13(b) for the range 0.5%-20%), but for higher percentage it showed either comparable or slightly worse performance.

[7]This image is produced by overlying 5 rotated versions of the image "wood.000", which is used to generate the image "wood45" in Figure 2.12(a), and taking its central part (of size 256×256).

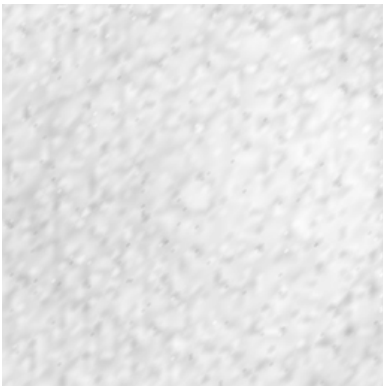


(a) Original image: "wood"



(b) PSNR of reconstructed image

(c) Reconstructed by tensor product



(d) Reconstructed by coset sum

Figure 2.13: An example of image with multiple directions. Comparison of approximation power of tensor product and coset sum for (a) original image "wood": 5-level-down decomposition and reconstruction using 3% largest coefficients. (c) The reconstructed image by tensor product, PSNR = 30.4 dB. (d) The reconstructed image by coset sum with $\Gamma' = \{(1,0),(0,1),(1,1)\}$, PSNR = 31.4 dB. (b) PSNR of reconstructed images over different percentage of retained coefficients (0.5%-20%).

We notice that in the reconstructed image in Figure 2.13(d) certain directions are pronounced more strongly than others despite that the original image in Figure 2.13(a) does not have that characteristic. This is due to the lack of rotational symmetry ([30]) of the coset sum refinable functions with the default coset representatives (cf. Figure 2.6 and 2.9(a)). A remedy for this can be obtained by choosing a set Γ' that gives (roughly)

equi-angled directions. For example, when $n = 2$, by setting $\Gamma' = \{(1,1), (-4,1), (1,-4)\}$, a rotational symmetry can be roughly achieved as it gives equi-angled directions $45°, 165°$, and $285°$ from the positive x-axis. However, in general it is not easy to overcome the lack of rotational symmetry of the coset sum refinable functions. For example, if we use the above non-default choice of Γ' for $\mathcal{C}_2[S_4]$ and $\mathcal{C}_2[U_4]$ in Example 4, our computation shows that the refinable function associated with $\mathcal{C}_2[S_4]$ is still in $L^2(\mathbb{R}^2)$, but the one associated with $\mathcal{C}_2[U_4]$ is not. Therefore obtaining coset sum refinable functions that are in $L^2(\mathbb{R}^n)$ with rotational symmetry may not be always possible even for the case of $n = 2$.

As a passing remark, we make a brief comment on comparison to the curvelet system [53], which is a state-of-the-art system for representing 2-D and 3-D data effectively using their geometric structure. Before presenting the image experiments using curvelets, we note that any comparison between the curvelet system and the coset sum wavelet system should be made with care as they are very different in nature. For one thing, the curvelet system is not a wavelet system constructed by using a method that works for any multi-D, which is our main interest in this paper. Besides, the curvelet system is highly redundant and its fast algorithm is slower than that of the tensor product and the coset sum system.

With all these in mind, we perform the curvelet transform to the above test images using the Matlab implementation (*fdct_wrapping.m* for decomposition and *ifdct_wrapping.m* for reconstruction) of 2-D discrete curvelets ([54]) in CurveLab Toolbox [55]. Even after fixing the decomposition level and the percentage of retained coefficients, there are still some parameters to be chosen in the curvelet codes, and the PSNR of reconstructed images is quite sensitive to the choice of these parameters. For reconstructed images using curvelet

(a) Reconstructed "part of lena" (b) Reconstructed "wood45" (c) Reconstructed "wood"

Figure 2.14: Reconstructed images by curvelets from a 5-level-down decomposition and retaining 3% largest coefficients, for original images "part of lena" (Figure 11(a)), "wood45" (Figure 12(a)) and "wood" (Figure 13(a)). Experiments are done by using the wrapping-based method ([54]) implemented in CurveLab Toolbox ([55]) with the default parameter setting. PSNR for reconstructed images are: (a) 25.4 dB (b) 27.7 dB and (c) 30.7 dB.

system with the default parameter setting, the PSNR is either between the PSNR of tensor product and that of coset sum ("wood45" and "wood"), or slightly lower than the PSNR of tensor product ("lena") (see Figure 2.14). In terms of the visual quality, the curvelet system is superior to the other two systems in both capturing different directional content and keeping rotational symmetry in an image (see Figure 2.14(c)), but it may add some strong directional artifacts to the reconstructed image (see Figure 2.14(a)(b)). We conclude that a complete comparison between the coset sum system and the curvelet system requires more thorough study on them.

2.5 Summary and outlook

In this paper we presented the coset sum as an alternative method to the tensor product in constructing decomposable multivariate refinement masks. The decomposable

refinement mask constructed by coset sum can be written as the sum, instead of the product, of the univariate refinement masks. We showed that the coset sum can provide many important features of the tensor product, such as preserving the biorthogonality of the univariate refinement masks and the availability of a wavelet system with fast algorithms.

Since the coset sum provides a way to obtain a pair of multivariate biorthogonal refinement masks, it can be combined with any method for finding wavelet masks to construct a (MRA-based biorthogonal) multivariate wavelet system. There has been only limited progress in a systematic construction of non-tensor based multivariate wavelet systems. The coset sum adds a new opportunity to this end.

By specifying wavelet masks as described in Section §2.4.1, we constructed a particular class of coset sum wavelet systems that can be associated with fast algorithms. Such algorithms are referred to as fast coset sum wavelet algorithms.

The fast tensor product wavelet algorithm has linear complexity, but the constant in the complexity bound increases as the spatial dimension increases. On the other hand, the constant in the (linear) complexity bound for the fast coset sum wavelet algorithm is independent of the dimension. Thus, when the spatial dimension is high, the coset sum wavelet algorithm can be *faster* than the tensor product wavelet algorithm.

Coset sum is not necessarily the only alternative to the tensor product. Rather, despite of its limitations in processing images, its existence with desirable features suggests that it may be worthwhile to develop and practice alternative methods to the tensor product for constructing multivariate wavelet systems.

2.6 Appendix

2.6.1 Proof of Theorem 1 in section §2.3.2

Proof of part (a)

Suppose H and h are the filters associated with masks R and $\mathcal{C}_n[R]$. If R is interpolatory, it is straightforward to show that $\mathcal{C}_n[R]$ is interpolatory. If $\mathcal{C}_n[R]$ is interpolatory, by (2.2), $h(0) = 1$, and $h(k) = 0$ if $k \in 2\mathbb{Z}^n\backslash 0$. Then by (3.7), $H(0) = 1$. Moreover, $H(K) = 0$ at all other even points, because if $H(K) \neq 0$ at some even point $K \in 2\mathbb{Z}\backslash 0$, then $h(k) = H(K) \neq 0$ at $k = K\nu \in 2\mathbb{Z}^n\backslash 0$, which contradicts to that $\mathcal{C}_n[R]$ is interpolatory. Therefore R is also interpolatory.

Proof of part (b)

Without loss of generality, we may assume $\tilde{R}$ is interpolatory. We want to show that, $\mathcal{C}_n[R]$ and $\mathcal{C}_n[\tilde{R}]$ are biorthogonal if and only if R and $\tilde{R}$ are biorthogonal.

Let $R^o := (R - R(\cdot + \pi))/2$ and $R^e := (R + R(\cdot + \pi))/2$ be the odd and even parts of R, respectively, and let $\tilde{R}^o$ be the odd part of $\tilde{R}$. Since $\tilde{R}$ is interpolatory, the even part of $\tilde{R}$ is the constant $1/2$. It is easy to check $\forall \omega_1 \in \mathbb{T}$

$$\overline{R^o(\omega_1)}\tilde{R}^o(\omega_1) = \frac{1}{2} - \frac{1}{2}\overline{R^e(\omega_1)}$$

$$\Longleftrightarrow R \text{ and } \tilde{R} \text{ are biorthogonal.}$$

Here, as before, the overline is used to denote the complex conjugate.

We will also need the following identities:

$$\sum_{\gamma \in \pi\Gamma} e^{-i\nu\cdot\gamma} = \begin{cases} 2^n, & \text{if } \nu = 0, \\ 0, & \text{if } \nu \in \Gamma'. \end{cases} \tag{2.15}$$

Then from the definition of the coset sum (cf. Definition 1, (2.6) and (2.7)), biorthogonal condition (2.1), and the above identities (2.15), we have

$$
\begin{aligned}
&\mathcal{C}_n[R] \text{ and } \mathcal{C}_n[\tilde{R}] \text{ are biorthogonal} \\
\Longleftrightarrow &\sum_{\gamma\in\pi\Gamma} (\overline{\mathcal{C}_n[R]}\mathcal{C}_n[\tilde{R}])(\omega+\gamma) = 1, \quad \forall \omega \in \mathbb{T}^n \\
\Longleftrightarrow &\left(\frac{1}{2^{n-1}}\right)^2 \sum_{\gamma\in\pi\Gamma} \left(-2^{n-1} + \sum_{\nu\in\Gamma} \overline{R((\omega+\gamma)\cdot\nu)}\right) \cdot \\
&\left(\frac{1}{2} + \sum_{\tilde{\nu}\in\Gamma'} \left(\tilde{R}((\omega+\gamma)\cdot\tilde{\nu}) - \frac{1}{2}\right)\right) = 1, \quad \forall \omega \in \mathbb{T}^n \\
\Longleftrightarrow &\sum_{\gamma\in\pi\Gamma} \left(1 - 2^{n-1} + \sum_{\nu\in\Gamma'} e^{i\gamma\cdot\nu}\overline{R^o(\omega\cdot\nu)} + \sum_{\nu\in\Gamma'} \overline{R^e(\omega\cdot\nu)}\right) \cdot \\
&\left(\frac{1}{2} + \sum_{\tilde{\nu}\in\Gamma'} e^{-i\gamma\cdot\tilde{\nu}}\tilde{R}^o(\omega\cdot\tilde{\nu})\right) = (2^{n-1})^2, \quad \forall \omega \in \mathbb{T}^n \\
\Longleftrightarrow &2^{n-1}(1-2^{n-1}) + 2^{n-1}\sum_{\nu\in\Gamma'} \overline{R^e(\omega\cdot\nu)} \\
&+2^n \sum_{\nu\in\Gamma'} \overline{R^o(\omega\cdot\nu)}\tilde{R}^o(\omega\cdot\nu) = (2^{n-1})^2, \quad \forall \omega \in \mathbb{T}^n \\
\Longleftrightarrow &\overline{R^o(\omega_1)}\tilde{R}^o(\omega_1) = \frac{1}{2} - \frac{1}{2}\overline{R^e(\omega_1)}, \quad \forall \omega_1 \in \mathbb{T}.
\end{aligned}
$$

Therefore, $\mathcal{C}_n[R]$ and $\mathcal{C}_n[\tilde{R}]$ are biorthogonal if and only if R and $\tilde{R}$ are biorthogonal.

Proof of part (c)

Let R be a univariate interpolatory refinement mask with accuracy number m. First let us prove the accuracy number of $\mathcal{C}_n[R]$ is at least m. Since R has accuracy number m,

$$(D^k R)(\pi) = 0, \ \forall 0 \le k \le m-1, \ \text{ and } \ (D^m R)(\pi) \neq 0. \tag{2.16}$$

Furthermore, since R is interpolatory, $1 - R(\omega) = R(\omega+\pi)$ holds for all $\omega \in \mathbb{T}$. Hence $(D^k(1-R))(0) = (D^k R)(\pi)$ for all $k \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$. Thus $1-R$ has a zero of order m at

the origin, i.e.

$$\begin{aligned} R(0) &= 1 \\ (D^k R)(0) &= 0, \quad \forall 1 \le k \le m-1 \\ (D^m R)(0) &\neq 0. \end{aligned} \tag{2.17}$$

Now consider the n-D refinement mask $\mathcal{C}_n[R]$. The accuracy number of $\mathcal{C}_n[R]$ is at least one, i.e. $\mathcal{C}_n[R](\gamma) = 0$, for all $\gamma \in \pi\Gamma'$. To see this, we need the dual identities of (2.15):

$$\sum_{\nu\in\Gamma} e^{-i\nu\cdot\gamma} = \begin{cases} 2^n, & \text{if } \gamma = 0, \\ 0, & \text{if } \gamma \in \pi\Gamma'. \end{cases} \tag{2.18}$$

From (2.18), we can read off

$$\#\{\nu \in \Gamma' : \gamma\cdot\nu \equiv \pi \,(\mathrm{mod}\, 2\pi\mathbb{Z})\} = 2^{n-1}, \tag{2.19}$$

for all $\gamma \in \pi\Gamma'$. In particular, the left-hand side of (2.19) is independent of γ. We then have for any $\gamma \in \pi\Gamma'$

$$\begin{aligned} 2^{n-1}\mathcal{C}_n[R](\gamma) &= -2^{n-1} + \sum_{\nu\in\Gamma} R(\gamma\cdot\nu) \\ &= 1 - 2^{n-1} + \sum_{\{\nu\in\Gamma':\gamma\cdot\nu\equiv 0\}} R(\gamma\cdot\nu) + \sum_{\{\nu\in\Gamma':\gamma\cdot\nu\equiv\pi\}} R(\gamma\cdot\nu) \\ &= 0, \end{aligned}$$

where $\equiv$ in the second line is used to denote congruence in modulo $2\pi\mathbb{Z}$, and the last equality is from the conditions $R(0) = 1$, $R(\pi) = 0$ and the identity (2.19). Furthermore, for all $\gamma \in \pi\Gamma'$ and for all $\mu \in \mathbb{N}_0^n$ with $1 \le |\mu| \le m-1$ ($|\mu| := \mu_1 + \cdots + \mu_n$)

$$\begin{aligned} (D^\mu \mathcal{C}_n[R])(\gamma) &= \frac{1}{2^{n-1}} \sum_{\nu\in\Gamma'} (D^\mu [R(\omega\cdot\nu)])_{|\omega=\gamma} \\ &= \frac{1}{2^{n-1}} \sum_{\nu\in\Gamma'} \left(\prod_{j=1}^n \nu_j^{\mu_j} \right) (D^{|\mu|} R)(\gamma\cdot\nu) = 0, \end{aligned}$$

where the last equality is from the identities (2.16) and (2.17). Therefore the accuracy number of $\mathcal{C}_n[R]$ is at least m.

Next we prove the accuracy number of $\mathcal{C}_n[R]$ is exactly m by contradiction. Suppose the accuracy number of $\mathcal{C}_n[R]$ is $m+l$ with $l \geq 1$. Then

$$(D^{\mu}\mathcal{C}_n[R])(\gamma) = 0,$$

$$\forall \gamma \in \pi\Gamma' \text{ and } \forall \mu \in \mathbb{N}_0^n \text{ with } 0 \leq |\mu| \leq m+l-1.$$

Since the univariate interpolatory R and the multivariate interpolatory $\mathcal{C}_n[R]$ are connected as follows:

$$R(\omega) = \mathcal{C}_n[R](\omega, 0, \cdots, 0), \quad \forall \omega \in \mathbb{T},$$

we have $(D^k R)(\pi) = D^{(k,0,\cdots,0)}\mathcal{C}_n[R](\pi, 0, \cdots, 0) = 0$ for all $0 \leq k \leq m+l-1$. Hence the accuracy number of R is at least $m+l$, which contradicts to the given assumption. Therefore the accuracy number of $\mathcal{C}_n[R]$ has to be m.

2.6.2 Proof of Theorem 2 in section §2.4.1

In this subsection we prove Theorem 2. In the proof we use the concepts of Compression-Alignment-Prediction (CAP) and Compression-Alignment-Modified-Prediction (CAMP) [56]. CAMP is a variant of CAP, and CAP is a generalization of the Laplacian pyramid [45]. In particular, CAP without alignment operator is the same as Laplacian pyramid. It is well known that Laplacian pyramid has a trivial reconstruction algorithm of reversing the steps in its decomposition algorithm. Both CAP and CAMP are originally designed for the redundant wavelet construction, and CAMP is introduced in order to achieve a better space localization than CAP.

Given $\tau := \mathcal{C}_n[S]$, $\tau^{\mathrm{d}} := \mathcal{C}_n[U]$ with interpolatory U, and $t_\nu(\omega) := e^{-i\nu\cdot\omega}\overline{U(\omega\cdot\nu+\pi)}$, $\omega \in \mathbb{T}^n$, $\nu \in \Gamma'$, we want to show that there exist dual wavelet masks $t^{\mathrm{d}}_\nu(\omega)$ such that $(\tau, (t_\nu)_{\nu\in\Gamma'})$ and $(\tau^{\mathrm{d}}, (t^{\mathrm{d}}_\nu)_{\nu\in\Gamma'})$ satisfy the MUEP conditions in (2.3).

To show this, first let us construct another pair of wavelet masks $(\tau_\nu)_{\nu\in\Gamma}$ and dual wavelet masks $(\tau^{\mathrm{d}}_\nu)_{\nu\in\Gamma}$, which we know for sure satisfy the MUEP conditions with τ and τ^{d}.

First extend the definition of t_ν by defining t_0:

$$t_\nu(\omega) := \begin{cases} \frac{1}{2}(1-\tau(\omega)), & \text{if } \nu = 0, \\ e^{-i\nu\cdot\omega}\overline{U(\omega\cdot\nu+\pi)}, & \text{if } \nu \in \Gamma'. \end{cases}$$

Then from [56] we know that

$$t_\nu(\omega) = 2^{\frac{n}{2}-1}\cdot t^{CAMP}_{-\nu}(\omega), \quad \nu\in\Gamma, \tag{2.20}$$

where t^{CAMP}_ν is the CAMPlet mask in Section §2.3 of [56].

Furthermore by comparing the CAPlet masks in Lemma 2.2 of [56] with the CAMPlet masks, it is easy to see that they are related as

$$t^{CAP}_\nu(\omega) - t^{CAMP}_\nu(\omega) = \begin{cases} 0, & \text{if } \nu = 0, \\ f_{-\nu}(\omega)t^{CAMP}_0(\omega), & \text{if } \nu\in\Gamma', \end{cases} \tag{2.21}$$

where $f_\nu(\omega) = e^{-i\nu\cdot\omega}\sum_{\gamma\in\pi\Gamma} e^{-i\nu\cdot\gamma}\overline{\tau^{\mathrm{d}}(\omega+\gamma)}$. Here it is necessary to point out that f_ν is π-periodic, i.e. $f_\nu(\omega+\gamma) = f_\nu(\omega)$, for any $\gamma\in\pi\Gamma$.

Now define $(\tau_\nu)_{\nu\in\Gamma}$

$$\tau_\nu(\omega) := 2^{\frac{n}{2}-1}t^{CAP}_{-\nu}(\omega), \quad \nu\in\Gamma. \tag{2.22}$$

Then since CAP without alignment operator is the same as Laplacian pyramid, and Lapla-

cian pyramid has a trivial reconstruction, we know that with

$$\tau_\nu^{\mathrm{d}}(\omega) := \begin{cases} 2^{1-n}, & \text{if } \nu = 0, \\ 2^{1-n} e^{-i\nu\cdot\omega}, & \text{if } \nu \in \Gamma', \end{cases}$$

$(\tau, (\tau_\nu)_{\nu\in\Gamma})$ and $(\tau^{\mathrm{d}}, (\tau_\nu^{\mathrm{d}})_{\nu\in\Gamma})$ satisfy the MUEP conditions.

Next, we start from the MUEP conditions of $(\tau, (\tau_\nu)_{\nu\in\Gamma})$ and $(\tau^{\mathrm{d}}, (\tau_\nu^{\mathrm{d}})_{\nu\in\Gamma})$ to find our dual wavelet masks t_ν^{d}. To do that, we need three more identities. The first one is a simple observation that can be obtained from (2.20), (2.21) and (2.22):

$$\tau_\nu(\omega) - t_\nu(\omega) = \begin{cases} 0, & \text{if } \nu = 0, \\ f_\nu(\omega) t_0(\omega), & \text{if } \nu \in \Gamma'. \end{cases} \tag{2.23}$$

The second one can be derived from the interpolatory property of τ^{d} and the identities (2.18):

$$\tau_0^{\mathrm{d}}(\omega) + \sum_{\nu\in\Gamma'} \overline{f_\nu(\omega)} \tau_\nu^{\mathrm{d}}(\omega) = 2\tau^{\mathrm{d}}(\omega). \tag{2.24}$$

After defining $g_\nu(\omega) := e^{-i\nu\cdot\omega} \sum_{\gamma\in\pi\Gamma} e^{-i\nu\cdot\gamma} \overline{\tau(\omega+\gamma)}$, the third identity:

$$t_0(\omega) + 2^{-n} \sum_{\nu\in\Gamma'} t_\nu(\omega) \overline{g_\nu(\omega)} = 0 \tag{2.25}$$

can be shown from the biorthogonality between τ and τ^{d} and the identities (2.18). Finally from the above identities (2.23), (2.24) and (2.25), with

$$\delta_{\gamma 0} := \begin{cases} 1, & \text{if } \gamma = 0, \\ 0, & \text{if } \gamma \in \pi\Gamma', \end{cases}$$

we get

$$\begin{aligned} &\delta_{\gamma 0} \\ =&\overline{\tau(\omega+\gamma)} \tau^{\mathrm{d}}(\omega) + \overline{\tau_0(\omega+\gamma)} \tau_0^{\mathrm{d}}(\omega) + \sum_{\nu\in\Gamma'} \overline{\tau_\nu(\omega+\gamma)} \tau_\nu^{\mathrm{d}}(\omega) \\ =&\overline{\tau(\omega+\gamma)} \tau^{\mathrm{d}}(\omega) + \overline{t_0(\omega+\gamma)} \tau_0^{\mathrm{d}}(\omega) \end{aligned}$$

$$
\begin{aligned}
&+\sum_{\nu\in\Gamma'}\overline{f_\nu(\omega+\gamma)t_0(\omega+\gamma)+t_\nu(\omega+\gamma)}\tau_\nu^{\mathrm{d}}(\omega)\\
=&\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega)\\
&+\overline{t_0(\omega+\gamma)}\left(\tau_0^{\mathrm{d}}(\omega)+\sum_{\nu\in\Gamma'}\overline{f_\nu(\omega+\gamma)}\tau_\nu^{\mathrm{d}}(\omega)\right)\\
&+\sum_{\nu\in\Gamma'}\overline{t_\nu(\omega+\gamma)}\tau_\nu^{\mathrm{d}}(\omega)\\
=&\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega)+\overline{t_0(\omega+\gamma)}2\tau^{\mathrm{d}}(\omega)+\sum_{\nu\in\Gamma'}\overline{t_\nu(\omega+\gamma)}\tau_\nu^{\mathrm{d}}(\omega)\\
=&\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega)\\
&+\left(\overline{t_0(\omega+\gamma)}+2^{-n}\sum_{\nu\in\Gamma'}\overline{t_\nu(\omega+\gamma)}g_\nu(\omega+\gamma)\right)2\tau^{\mathrm{d}}(\omega)\\
&-2^{-n}\sum_{\nu\in\Gamma'}\overline{t_\nu(\omega+\gamma)}g_\nu(\omega+\gamma)2\tau^{\mathrm{d}}(\omega)\\
&+\sum_{\nu\in\Gamma'}\overline{t_\nu(\omega+\gamma)}\tau_\nu^{\mathrm{d}}(\omega)\\
=&\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega)-2^{1-n}\sum_{\nu\in\Gamma'}\overline{t_\nu(\omega+\gamma)}g_\nu(\omega+\gamma)\tau^{\mathrm{d}}(\omega)\\
&+\sum_{\nu\in\Gamma'}\overline{t_\nu(\omega+\gamma)}\tau_\nu^{\mathrm{d}}(\omega)\\
=&\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega)\\
&+\sum_{\nu\in\Gamma'}\overline{t_\nu(\omega+\gamma)}\left(-2^{1-n}g_\nu(\omega)\tau^{\mathrm{d}}(\omega)+\tau_\nu^{\mathrm{d}}(\omega)\right).
\end{aligned}
$$

Therefore, by letting $t_\nu^{\mathrm{d}} := -2^{1-n}g_\nu\tau^{\mathrm{d}}+\tau_\nu^{\mathrm{d}}$, we find the dual wavelet masks t_ν^{d}, $\nu\in\Gamma'$, such that $(\tau,(t_\nu)_{\nu\in\Gamma'})$ and $(\tau^{\mathrm{d}},(t_\nu^{\mathrm{d}})_{\nu\in\Gamma'})$ satisfy the MUEP conditions.

2.6.3 Proof of the identity in Step (iii) of the coset sum algorithm in section §2.4.2

In this subsection we verify the identity in Step (iii) of Reconstruction Algorithm in Section §2.4.2. We use the same notation as in the algorithm. In particular, G and H are univariate refinement filters associated with biorthogonal refinement masks S and U,

respectively, and H is interpolatory.

From Step `(i)` of the algorithm in Section §2.4.2, we know that, with $a_G = 2^n - (2^n-1)(2-G(0))$,

$$\begin{aligned} a_G\, y_j(2k) =& 2^n y_{j-1}(k) - \sum_{\nu\in\Gamma'}\sum_{L\in\mathbb{Z}\setminus 0} G(L) y_j(2k+L\nu) \\ =& 2^n y_{j-1}(k) - \sum_{\nu\in\Gamma'}\sum_{L\equiv 1} G(L) y_j(2k+L\nu) \\ & - \sum_{\nu\in\Gamma'}\sum_{L\equiv 0, L\neq 0} G(L) y_j(2k+L\nu) \end{aligned} \tag{2.26}$$

where $\equiv$ is used to denote congruence in modulo $2\mathbb{Z}$. Since the masks S and U are biorthogonal, from (2.1) and the connection between the filter and the mask, it is easy to see that the associated filters G and H satisfy the following condition:

$$\sum_{m\in\mathbb{Z}} G(L+m)H(m) = \begin{cases} 0, & \text{if } L\equiv 0, L\neq 0, \\ \\ 2, & \text{if } L = 0. \end{cases}$$

Combining this with the fact that H is interpolatory leads to

$$\sum_{m\equiv 1} G(L+m)H(m) = \begin{cases} 0 - G(L), & \text{if } L\equiv 0, L\neq 0, \\ \\ 2 - G(0), & \text{if } L = 0. \end{cases}$$

From this and the change of variables, we see that

$$\begin{aligned} & \sum_{\nu\in\Gamma'}\sum_{L\equiv 0, L\neq 0} G(L) y_j(2k+L\nu) \\ =& \sum_{\nu\in\Gamma'}\sum_{L\equiv 0, L\neq 0} \left(0 - \sum_{m\equiv 1} G(L+m)H(m)\right) y_j(2k+L\nu) \\ =& -\sum_{\nu\in\Gamma'}\sum_{m\equiv 1}\sum_{L\equiv 0} G(L+m)H(m) y_j(2k+L\nu) \\ & + \sum_{\nu\in\Gamma'}\sum_{m\equiv 1} G(m)H(m) y_j(2k) \\ =& -\sum_{\nu\in\Gamma'}\sum_{m\equiv 1}\sum_{n\equiv 1} G(n)H(m) y_j(2k+(n-m)\nu) \\ & + (2^n-1)(2-G(0)) y_j(2k) \end{aligned}$$

$$\begin{aligned}=&-\sum_{\nu\in\Gamma'}\sum_{L\equiv 1}\sum_{m\equiv 1} G(L)H(m)y_j(2k+(L-m)\nu)\\&+(2^n-1)(2-G(0))y_j(2k)\end{aligned}$$

By substituting this result to (2.26) and solving for $y_j(2k)$, we obtain

$$y_j(2k) = y_{j-1}(k) - \frac{1}{2^{n-1}}\sum_{\nu\in\Gamma'}\sum_{L\in\mathbb{Z}} G(2L+1)w_{\nu,j-1}(k+L\nu)$$

as desired.

Chapter 3

Prime Coset Sum: A Systematic Method for Designing Multi-D Wavelet Filter Banks with Fast Algorithms

3.1 Preliminaries

3.1.1 Introduction

Wavelet representation has been one of the most popular data representations in the last two decades. Wavelet filter banks, which can lead to wavelet systems in $L_2(\mathbb{R}^n)$ under some well-understood constraints, has been widely used in signal processing applications. In order to obtain wavelet representation for multi-dimensional (multi-D) data, one

needs multi-D wavelets. Tensor product is the most common method for constructing multi-D wavelets, and the resulting wavelets are typically referred to as the separable wavelets. However, the separable wavelets constitute only a small portion of multi-D wavelets, and they have some unavoidable limitations. One of the limitations of tensor-based wavelets is that the resulting multi-D filters have dense supports. It is well known that the fast algorithms associated with tensor-based wavelets have a complexity constant (cf. Section §3.3.2 for the definition of complexity constant) that increases *linearly* with the spatial dimension n. While this complexity may be satisfactory for many of the regular 2-D image processing, it can pose a problem when dealing with large volume data such as medical images in [57], Geographic Information Systems images in [58] and seismic data in [59]. Moreover, tensor-based discrete wavelet transform is memory consuming and cannot be used to directly obtain the target subband signals, due to its dependent subband decomposition process [60]. Motivated by the aforementioned drawbacks, much work has been done to improve the implementation of tensor-based wavelets [61–64]. There have been many researches on non-tensor-based multi-D wavelet constructions too [7–20, 49, 53, 65–72]. However, most of these methods work only for low dimensions or have additional constraints on the lowpass filters. Furthermore, most of them are not associated with fast algorithms, preventing them from being widely used in practice.

Recently, the authors introduced a new method called coset sum for constructing non-tensor-based multi-D wavelets in [4]. There it was shown that the resulting wavelets are associated with fast algorithms whose complexity constant does not increase as the spatial dimension increases. It was also shown there that many features of tensor product that

makes it attractive in wavelet construction still hold true for coset sum.

However, similar to the tensor product method, coset sum also assumes the dyadic dilation. We recall that the $n \times n$ matrix Λ is called a *dilation matrix* if it is an integer matrix whose spectrum lies outside the closed unit disc. It determines the exact way of how downsampling and upsampling are performed in wavelets or wavelet filter banks. The dilation is called *scalar* if the dilation matrix is the scalar multiple of the identity matrix, i.e., $\Lambda = \lambda \mathtt{I}_n$ with $\lambda \geq 2$ an integer. In particular, it is called *dyadic* if $\Lambda = 2\mathtt{I}_n$ and *prime* if $\Lambda = p\mathtt{I}_n$ with p a prime number. Wavelets with dyadic dilation are referred to as dyadic wavelets. Dyadic wavelets are the standard and traditional types of wavelets, however they are not suitable for all applications. For non-dyadic frequency divisions [73], non-dyadic scale ratios [74], or flexible decompositions of the data [75], non-dyadic wavelets are more suitable.

In this paper, we show that we can generalize the coset sum in the sense that multi-D wavelet filter banks with fast algorithms can be constructed for any prime dilation $p\mathtt{I}_n$. We also show that the complexity constant for our fast algorithms with prime dilation $p\mathtt{I}_n$ is independent of the spatial dimension.

The organization of this paper is as follows. The rest of Section §3.1 is a brief review of some relevant concepts including the coset sum method. In Section §3.2 we discuss a possible generalization of the coset sum, which we call prime coset sum, together with its properties. In Section §3.3 we present a new method to construct multi-D wavelet filter banks based on the prime coset sum refinement masks and show that they are associated with fast algorithms. Section §3.4 is a summary of our results. Some technical proofs and

details in this paper are placed in Appendix §3.5.

3.1.2 Notation and basic concepts

Let Λ be a dilation matrix and let $q := |\det \Lambda|$. In the multiresolution analysis [1] setting, the (compactly supported) scaling or refinable function ϕ (with dilation Λ) satisfies the following refinement relation:

$$\phi(\cdot) = \sum_{k\in\mathbb{Z}^n} h_\phi(k)\phi(\Lambda\cdot - k), \tag{3.1}$$

where $h_\phi : \mathbb{Z}^n \to \mathbb{R}$ is the associated finitely supported filter with dilation Λ.

A mask *associated with* a finitely supported filter $h : \mathbb{Z}^n \to \mathbb{R}$ is a Laurent trigonometric polynomial defined as

$$\tau(\omega) := \frac{1}{q}\sum_{k\in\mathbb{Z}^n} h(k)e^{-ik\cdot\omega} =: \widehat{h}(\omega),$$

for any $\omega \in \mathbb{T}^n := [-\pi,\pi]^n$. That is, $\tau = \widehat{h}$ is the Fourier transform of the filter h, up to a normalization. Throughout this paper, we use $\widehat{a}$ to denote this Fourier transform of a.

By taking the Fourier transform of (3.1), the refinement relation can be recast as

$$\widehat{\phi}(\Lambda^*\omega) = \tau(\omega)\widehat{\phi}(\omega), \quad \forall\omega\in\mathbb{T}^n,$$

where τ is the mask associated with h_ϕ, and $*$ is used to denote the conjugate transpose of a matrix, hence Λ^* is the same as Λ^T, the transpose of Λ, in this case.

A mask τ with $\tau(0) = 0$ is typically referred to as a *wavelet mask*. In this paper, we use the normalization of the mask so that a mask with $\tau(0) = 1$ is referred to as a *refinement mask*. This is equivalent to $\sum_{k\in\mathbb{Z}^n} h(k) = q$, which is our normalization for a

filter to be *lowpass*. A refinement mask τ is called *interpolatory* if

$$\sum_{\gamma\in\Gamma^*}\tau(\omega+\gamma)=1,$$

for any $\omega\in\mathbb{T}^n$, where Γ^* is a complete set of representatives of the distinct cosets of $2\pi(((\Lambda^*)^{-1}\mathbb{Z}^n)/\mathbb{Z}^n)$ containing 0. For example, for the scalar dilation with λ, the set $\frac{2\pi}{\lambda}\{0,1,\cdots,\lambda-1\}^n$ can be used for Γ^*. We note that τ is interpolatory if and only if its corresponding filter h satisfies

$$h(k)=\begin{cases}1, & \text{if } k=0,\\ \\ 0, & \text{if } k\in\Lambda\mathbb{Z}^n\backslash 0.\end{cases}\tag{3.2}$$

The order of zeros of τ at $\gamma\in\Gamma^*\backslash 0$ is called the accuracy number of τ. Throughout this paper, we assume that all refinement masks have at least accuracy number one. The order of zeros of τ at the origin is called the number of vanishing moments of τ. Thus a mask is a wavelet mask if and only if it has at least one vanishing moment. The order of zeros of $1-\tau$ at the origin is called the flatness number of τ. Thus a mask is a refinement mask if and only if it has at least flatness number one. Throughout this paper, we use the accuracy number, the number of vanishing moments, and the flatness number both for a mask and for the filter associated with it.

Two refinement masks τ and τ^{d} are called *biorthogonal* if

$$\sum_{\gamma\in\Gamma^*}(\overline{\tau}\tau^{\mathrm{d}})(\omega+\gamma)=1,$$

for any $\omega\in\mathbb{T}^n$. Here and below, the overline is used to denote the complex conjugate. For the corresponding filters h and g of τ and τ^{d}, respectively, the biorthogonality condition becomes

$$\sum_{k\in\mathbb{Z}^n}h(k)g(k+\Lambda l)=q\delta_{l,0}=\begin{cases}q, & \text{if } l=0,\\ \\ 0, & \text{if } l\in\mathbb{Z}^n\backslash 0.\end{cases}$$

For a pair of biorthogonal refinement masks τ and τ^{d} and wavelet masks t_j and t_j^{d}, $j = 1, \ldots, q-1$, we refer to $(\tau, (t_j)_{j=1,\ldots,q-1})$ and $(\tau^{\mathrm{d}}, (t_j^{\mathrm{d}})_{j=1,\ldots,q-1})$ as the *combined biorthogonal masks* if they satisfy the following condition: for every $\omega \in \mathbb{T}^n$,

$$\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega) + \sum_{j=1}^{q-1} \overline{t_j(\omega+\gamma)} t_j^{\mathrm{d}}(\omega) = \delta_{\gamma,0} = \begin{cases} 1, & \text{if } \gamma = 0, \\ 0, & \text{if } \gamma \in \Gamma^* \backslash 0. \end{cases} \tag{3.3}$$

It is well known that the combined biorthogonal masks can give rise to a biorthogonal wavelet system in $L_2(\mathbb{R}^n)$ (see, for example, [33]).

A filter bank is a finite set of filters. We consider only the filter banks that are non-redundant with the perfect reconstruction property [3]. A (non-redundant) filter bank consists of analysis bank and synthesis bank, which are collections of $q = |\det \Lambda|$ filters linked by downsampling and upsampling operators, respectively, associated with the dilation matrix Λ. The analysis bank splits the input signal into q signals typically called subband signals using a parallel set of bandpass filters. The synthesis bank reconstructs the original data from q subband signals. We are interested in the *wavelet filter bank* for which each of analysis and synthesis banks has exactly one lowpass filter and the rest of them are all highpass filters. We recall that a filter h is *highpass* if the associated mask is a wavelet mask, i.e. $\sum_{k\in\mathbb{Z}^n} h(k) = 0$. The filters associated with the combined biorthogonal masks constitute a wavelet filter bank. Furthermore, it is well known that the minimum of accuracy numbers of lowpass filters in a wavelet filter bank provides a lower bound for the number of vanishing moments of the highpass filters in the same wavelet filter bank [36].

3.1.3 Multi-D wavelet construction methods: tensor product and coset sum

When $q = |\det \Lambda|$ is large, in general, it is not easy to find the combined biorthogonal masks $(\tau, (t_j)_{j=1,\dots,q-1})$ and $(\tau^{\mathrm{d}}, (t_j^{\mathrm{d}})_{j=1,\dots,q-1})$. However, if the dilation is dyadic (i.e. $\Lambda = 2\mathtt{I}_n$ and $q = 2^n$) and the spatial dimension n satisfies $n \geq 2$, then the well-known tensor product and more recent coset sum can be used. Below we provide a brief review of these methods.

We recall that the n-D tensor product mask from n (possibly distinct) 1-D masks $R_1, R_2, \dots, R_n$ is defined as, for $\omega = (\omega_1, \omega_2, \dots, \omega_n) \in \mathbb{T}^n$,

$$\mathcal{T}_n[R_1, R_2, \dots, R_n](\omega) := R_1(\omega_1) R_2(\omega_2) \cdots R_n(\omega_n).$$

Then starting from 1-D combined biorthogonal masks (S_0, S_1) and (U_0, U_1) with dyadic dilation, one can construct n-D combined biorthogonal masks with dyadic dilation by setting the n-D biorthogonal refinement masks as

$$\tau := \mathcal{T}_n[S_0, S_0, \dots, S_0], \quad \tau^{\mathrm{d}} := \mathcal{T}_n[U_0, U_0, \dots, U_0],$$

and the n-D wavelet masks t_ν, t_ν^{d}, $\nu = (\nu_1, \nu_2, \dots, \nu_n) \in \{0,1\}^n \backslash 0$, as

$$t_\nu = \mathcal{T}_n[S_{\nu_1}, S_{\nu_2}, \dots, S_{\nu_n}], \quad t_\nu^{\mathrm{d}} = \mathcal{T}_n[U_{\nu_1}, U_{\nu_2}, \dots, U_{\nu_n}].$$

It is well known that the above tensor product method has many advantages: 1) it preserves the interpolatory property and the accuracy number of 1-D refinement masks; 2) it also preserves the biorthogonality between two refinement masks; and 3) the resulting separable wavelets are associated with fast algorithms (cf. Section §3.3.2). However, as

discussed in Section §3.1.1, the limitations of the separable wavelets constructed from the tensor product are also prominent.

A new alternative method called coset sum ([4]) for constructing n-D dyadic refinement masks from 1-D dyadic refinement masks is recently proposed. The coset sum refinement mask $\mathcal{C}_n[R]$ for a 1-D dyadic refinement mask R is defined as:

$$\mathcal{C}_n[R](\omega) := \frac{1}{2^{n-1}} \left(1 - 2^{n-1} + \sum_{\nu \in \{0,1\}^n \backslash 0} R(\omega \cdot \nu) \right), \quad \omega \in \mathbb{T}^n.$$

The following results about coset sum refinement masks and coset sum wavelet systems have been proved in [4].

Result 1 *Let $\mathcal{C}_n$ be the coset sum, and let R and $\tilde{R}$ be univariate dyadic refinement masks.*

(a) $\mathcal{C}_n[R]$ is interpolatory if and only if R is interpolatory.

(b) Suppose that one of R and $\tilde{R}$ is interpolatory. Then $\mathcal{C}_n[R]$ and $\mathcal{C}_n[\tilde{R}]$ are biorthogonal if and only if R and $\tilde{R}$ are biorthogonal.

(c) Suppose that R is interpolatory. Then $\mathcal{C}_n[R]$ and R have the same accuracy number. ■

Result 2 *Suppose that S and U are 1-D biorthogonal dyadic refinement masks, and that U is interpolatory. Define n-D biorthogonal refinement masks as*

$$\tau := \mathcal{C}_n[S], \quad \tau^{\mathrm{d}} := \mathcal{C}_n[U],$$

and n-D wavelet masks t_ν, $\nu \in \{0,1\}^n \backslash 0$, as

$$t_\nu(\omega) = e^{-i\omega\cdot\nu} \overline{U(\omega \cdot \nu + \pi)}, \quad \omega \in \mathbb{T}^n. \tag{3.4}$$

Then there exist t^{d}_ν, $\nu \in \{0,1\}^n \backslash 0$, such that $(\tau, (t_\nu)_{\nu \in \{0,1\}^n \backslash 0})$ and $(\tau^{\mathrm{d}}, (t^{\mathrm{d}}_\nu)_{\nu \in \{0,1\}^n \backslash 0})$ are n-D combined biorthogonal masks with dyadic dilation. ■

As we can see above, the coset sum and the tensor product method share many useful properties. In addition, the coset sum wavelets can overcome some of the limitations of the separable wavelets. For example, attributed to the smaller *supports* (number of nonzero entries) of the resulting multi-D filters, as well as the special structure of the filters, the coset sum can be associated with fast algorithms whose complexity constant does not increase with the spatial dimension. Therefore, in higher dimension, coset sum fast algorithms can be much faster than the tensor product fast algorithms. For more details about the coset sum including its comparison with the tensor product, we refer to [4].

3.2 Prime coset sum

Since coset sum has many attractive properties including fast algorithms, which can be much faster than the existing tensor product fast algorithms, in this section, we try to extend the coset sum method to non-dyadic scalar dilations. For the usefulness of non-dyadic dilation wavelets, we refer to the discussion in Section §3.1.1. The following simple lemma plays an important role in our generalization of coset sum.

Lemma 1 *Let p be a prime number, and let Γ and Γ^* be the complete set of representatives of the distinct cosets of $\mathbb{Z}^n/p\mathbb{Z}^n$ and $2\pi((p^{-1}\mathbb{Z}^n)/\mathbb{Z}^n)$, respectively, containing 0. Then for every $\gamma \in \Gamma^* \backslash 0$, we have*

$$\#\{\nu \in \Gamma : \ \gamma \cdot \nu \equiv 0 \,(\mathrm{mod}\, 2\pi\mathbb{Z})\} = p^{n-1}. \qquad \blacksquare$$

Remark 1 *A special case of Lemma 1 for $p = 2$ is used for the coset sum (cf. (19) in [4]).* $\blacksquare$

Remark 2 *In general, Lemma 1 does not hold true if p is not a prime number. For example, when $p = 4$ and $n = 1$, we can take $\Gamma = \{0, 1, 2, 3\}$ and $\Gamma^* \backslash 0 = \{\frac{2\pi}{4}, \frac{4\pi}{4}, \frac{6\pi}{4}\}$. Then, it is easy to see that if $\gamma = \frac{2\pi}{4}$ or $\gamma = \frac{6\pi}{4}$, then the cardinality of the set $Z_\gamma := \{\nu \in \Gamma : \ \gamma \cdot \nu \equiv 0 \,(\mathrm{mod}\, 2\pi\mathbb{Z})\}$ is 1 (in fact, $Z_\gamma = \{0\}$ in both cases), whereas if $\gamma = \frac{4\pi}{4}$, then $Z_\gamma = \{0, 2\}$ and hence its cardinality is 2. As we will see below, a crucial step in our proof of the lemma is the fact that $\mathbb{Z}/p\mathbb{Z}$ is a finite field for a prime number p, which does not hold true anymore if p is not a prime number.* ■

Proof 1 (Proof of Lemma 1) *First of all, we claim that, without lose of generality, we may assume $\Gamma = \{0, 1, \cdots, p-1\}^n$ and $\Gamma^* = \frac{2\pi}{p}\{0, 1, \cdots, p-1\}^n$. This is because for any other $\tilde{\Gamma}$ and $\tilde{\Gamma}^*$, there is a one-to-one correspondence between the elements of $\tilde{\Gamma}$ and Γ, and between the elements of $\tilde{\Gamma}^*$ and Γ^*. To be more specific, for any other $\tilde{\Gamma}$ and $\tilde{\Gamma}^*$, and for any $\tilde{\nu} \in \tilde{\Gamma}$ and $\tilde{\gamma} \in \tilde{\Gamma}^* \backslash 0$, there exist unique $\nu \in \Gamma$ and $\gamma \in \Gamma^* \backslash 0$ such that*

$$\nu \equiv \tilde{\nu} \,(\mathrm{mod}\, p\mathbb{Z}^n), \quad \frac{p}{2\pi}\gamma \equiv \frac{p}{2\pi}\tilde{\gamma} \,(\mathrm{mod}\, p\mathbb{Z}^n),$$

and vice versa. Therefore, $\tilde{\gamma} \cdot \tilde{\nu} \equiv \gamma \cdot \nu \,(\mathrm{mod}\, 2\pi\mathbb{Z})$. Hence the cardinality of the set $\{\nu \in \Gamma : \ \gamma \cdot \nu \equiv 0 \,(\mathrm{mod}\, 2\pi\mathbb{Z})\}$ is the same as the cardinality of the set $\{\tilde{\nu} \in \tilde{\Gamma} : \ \tilde{\gamma} \cdot \tilde{\nu} \equiv 0 \,(\mathrm{mod}\, 2\pi\mathbb{Z})\}$.

Now for any $\gamma \in \Gamma^* \backslash 0 = \frac{2\pi}{p}\{0, 1, \cdots, p-1\}^n \backslash 0$, and $\nu \in \Gamma = \{0, 1, \cdots, p-1\}^n$, we let $\mu := \frac{p}{2\pi}\gamma$, and let μ_i and ν_i, $i = 1, \ldots, n$, be the i-th component of μ and ν. Then both μ_i and ν_i lie in the set $\{0, 1, \cdots, p-1\}$. Since $\gamma \neq 0$, at least one of μ_i's is not 0. Without loss of generality, we may assume $\mu_n \neq 0$. Furthermore, $\gamma \cdot \nu \equiv 0 \,(\mathrm{mod}\, 2\pi\mathbb{Z})$ if and only if $\mu_1\nu_1 + \cdots + \mu_n\nu_n \equiv 0 \,(\mathrm{mod}\, p\mathbb{Z})$.

For any $\gamma \in \Gamma^* \backslash 0$, and any $\nu_i \in \{0, 1, \cdots, p-1\}$, $i = 1, \ldots, n-1$, let $k \in$

$\{0, 1, \cdots, p-1\}$ *satisfy*

$$\mu_1\nu_1 + \cdots + \mu_{n-1}\nu_{n-1} \equiv k \,(\mathrm{mod}\, p\mathbb{Z}).$$

Since $\mathbb{Z}/p\mathbb{Z}$ *is a finite field for a prime number* p*, there exists a unique multiplicative inverse* $\rho(\mu_n) \in \{1, \cdots, p-1\}$ *with* $\mu_n\rho(\mu_n) \equiv 1 \,(\mathrm{mod}\, p\mathbb{Z})$*. Then there exists a unique* $\nu_n \in \{0, 1, \cdots, p-1\}$ *satisfies*

$$\nu_n \equiv (-k)\rho(\mu_n) \,(\mathrm{mod}\, p\mathbb{Z}).$$

Thus

$$\mu_1\nu_1 + \cdots + \mu_{n-1}\nu_{n-1} + \mu_n\nu_n \equiv k + \mu_n(-k)\rho(\mu_n) \equiv 0 \,(\mathrm{mod}\, p\mathbb{Z}).$$

Since there are p^{n-1} *different choices for* $\nu_1, \nu_2, \cdots, \nu_{n-1}$*, for any* $\gamma \in \Gamma^* \backslash 0$*, we have*

$$\#\{\nu \in \Gamma : \ \gamma \cdot \nu \equiv 0 \,(\mathrm{mod}\, 2\pi\mathbb{Z})\} = p^{n-1}. \qquad \blacksquare$$

With Lemma 1 in hand, we define a particular generalization of coset sum for the prime dilation $\Lambda = p\mathtt{I}_n$, where $p \geq 2$ is a prime number. Let Γ and Γ^* be defined as in Lemma 1. For example, $\Gamma = \{0, 1, \cdots, p-1\}^n$ and $\Gamma^* = \frac{2\pi}{p}\{0, 1, \cdots, p-1\}^n$ can be used.

Motivated by the definition of the original coset sum $\mathcal{C}_n$ (cf. Section §3.1.3), we consider a generalized coset sum $\mathcal{C}_{n,p}$ of the form

$$\mathcal{C}_{n,p}[R](\omega) = A\left(B + \sum_{\nu \in \Gamma'} R(\omega \cdot \nu)\right),$$

where $\Gamma' := \Gamma \backslash 0$, and A and B are constants that will be determined soon. To pin down the constants A and B, we impose two conditions that we consider *natural* on the map $\mathcal{C}_{n,p}$. Firstly, we require $\mathcal{C}_{n,p}$ to map a 1-D refinement mask with dilation p to an n-D refinement

mask with dilation $p\mathtt{I}_n$. That is, we want $\mathcal{C}_{n,p}[R](0) = 1$ whenever $R(0) = 1$. From this we get the equation

$$B + p^n - 1 = \frac{1}{A}. \tag{3.5}$$

Secondly, we require the accuracy number of $\mathcal{C}_{n,p}[R]$ to be at least one whenever the accuracy number of the 1-D refinement mask R is at least one. That is, we want, for any $\gamma \in \Gamma^* \backslash 0$,

$$0 = \mathcal{C}_{n,p}[R](\gamma) = A\left(B + \sum_{\{\nu \in \Gamma', \gamma\cdot\nu \equiv 0\}} R(0)\right) = A\left(B + p^{n-1} - 1\right),$$

where the last equality is due to Lemma 1. This gives the equation

$$B + (p^{n-1} - 1) = 0. \tag{3.6}$$

By solving A and B that satisfy (3.5) and (3.6) simultaneously, we reach the following definition of a generalized coset sum for prime dilations.

Definition 2 *Let p be a prime number. We define the prime coset sum $\mathcal{C}_{n,p}$ that maps a 1-D refinement mask R with dilation p to an n-D refinement mask $\mathcal{C}_{n,p}[R]$ with dilation $p\mathtt{I}_n$ as follows: for any $\omega \in \mathbb{T}^n$,*

$$\mathcal{C}_{n,p}[R](\omega) := \frac{1}{(p-1)p^{n-1}} \left(1 - p^{n-1} + \sum_{\nu \in \Gamma'} R(\omega \cdot \nu)\right),$$

where $\Gamma' = \Gamma \backslash 0$. ■

Remark 3 *We refer to the refinement mask obtained by $\mathcal{C}_{n,p}$ as the prime coset sum refinement mask. We notice that the prime coset sum $\mathcal{C}_{n,p}$ with $p = 2$ reduces to the original coset sum $\mathcal{C}_n$ for dyadic dilation, i.e. $\mathcal{C}_{n,2} = \mathcal{C}_n$ (cf. Section §3.1.3 for the choice of $\Gamma = \{0, 1\}^n$ and [4] for more general choice of Γ).* ■

$$1 \quad \mathbf{1} \quad 1 \quad \longrightarrow \quad \begin{matrix} 1 & 1 & 1 \\ 1 & \mathbf{1} & 1 \\ 1 & 1 & 1 \end{matrix}$$

Figure 3.1: Construction of centered 2-D Haar lowpass filter with dilation 3 using prime coset sum (cf. Example 1)[2]

Let H be the 1-D lowpass filter associated with the 1-D refinement mask R. Let h be the n-D lowpass filter associated with the n-D refinement mask $\mathcal{C}_{n,p}[R]$. We refer to such a filter h as the *prime coset sum lowpass filter*. For any nonzero $k \in \mathbb{Z}^n$, we define a set W_k as $W_k := \{l \in \mathbb{Z}\backslash 0 : k = l\nu \text{ for some } \nu \in \Gamma'\}$. Then the n-D prime coset sum lowpass filter h can be written in terms of the 1-D lowpass filter H as follows:

$$h(k) = \begin{cases} \frac{1}{p-1}(p - p^n + (p^n - 1)H(0)), & \text{if } k = 0, \\ \frac{1}{p-1}\sum_{l \in W_k} H(l), & \text{if } k \neq 0. \end{cases} \tag{3.7}$$

Now we give a simple example to show the construction of multi-D prime coset sum lowpass filters.

Example 1 (Centered 2-D Haar lowpass filter with dilation 3). *Consider the centered 1-D Haar lowpass filter with dilation 3*

$$H(K) = \begin{cases} 1, & \textit{if } K = 0 \textit{ or } K = \pm 1, \\ 0, & \textit{otherwise.} \end{cases}$$

Let us take $\Gamma = \{-1, 0, 1\}^2 = \{(0,0), \pm(1,0), \pm(0,1), \pm(1,1), \pm(1,-1)\}$. *Then it is easy to check that the 2-D prime coset sum lowpass filter constructed from the 1-D centered Haar is*

$$h(k) = \begin{cases} 1, & \textit{if } k = (0,0),\ k = \pm(1,0),\ k = \pm(0,1),\ k = \pm(1,-1) \textit{ or } k = \pm(-1,1), \\ 0, & \textit{otherwise.} \end{cases}$$

[2]Bold-faced number indicates that it is at the origin. This figure is also given out in [76].

Figure 3.1 shows the 1*-D filter* H *and the resulting* 2*-D filter* h*.* ■

Some of the properties of the original coset sum (cf. Section §3.1.3) still hold true for the generalized prime coset sum.

Lemma 2 *Let* $\mathcal{C}_{n,p}$ *be the prime coset sum, and* R *be a univariate refinement mask with dilation* p*. If* R *is interpolatory, then* $\mathcal{C}_{n,p}[R]$ *is interpolatory.* ■

Proof 2 *See Appendix §3.5.1.* ■

Lemma 3 *Let* $\mathcal{C}_{n,p}$ *be the prime coset sum,* R *be a univariate refinement mask with dilation* p*, and let* m_1 *and* m_2 *be positive integers. Suppose that* R *has* m_1 *accuracy and* m_2 *flatness. Then* $\mathcal{C}_{n,p}[R]$ *has at least* $\min\{m_1, m_2\}$ *accuracy.* ■

Proof 3 *See Appendix §3.5.2. Similar arguments to the ones given in [4] are used in our proof.* ■

Remark 4 *If* R *is interpolatory, then* $m_1 = m_2$*. Hence, the above lemma says that, when* R *is interpolatory, the accuracy number of* $\mathcal{C}_{n,p}[R]$ *is at least as many as the accuracy number of* R*. For the case of the original coset sum with dyadic dilation, the accuracy number of* $\mathcal{C}_n[R]$ *is exactly the same as the accuracy number of* R *when* R *is interpolatory (cf. Result 1(c)). We do not yet know whether this result would hold true for the prime coset sum in general.* ■

Lemma 4 *Let* $\mathcal{C}_{n,p}$ *be the prime coset sum, and* R *be a univariate refinement mask with dilation* p*. Then the flatness number of* $\mathcal{C}_{n,p}[R]$ *is at least the flatness number of* R*.* ■

We omit the proof of Lemma 4 as it is a simple variant of our proof of Lemma 3.

Unlike the original coset sum with dyadic dilation (cf. Result 1(b)), in general, the prime coset sum does not preserve the biorthogonality of 1-D refinement masks when $p > 2$, even if one of them is interpolatory. Let us look at two examples to this end. Both of them are related with the Haar refinement masks with dilation 3.

Example 2 (Centered 2-D Haar refinement mask with dilation 3). *Let us consider the centered 1-D Haar refinement mask as in Example 1:*

$$\frac{1}{3}\left(e^{i\omega} + 1 + e^{-i\omega}\right).$$

Then the above mask has dilation 3 and it is associated with the refinable function $\phi = \chi_{[-1/2,1/2]}$. If we define both R and $\tilde{R}$ to be this centered 1-D Haar refinement mask with dilation 3, then they are interpolatory and biorthogonal with one accuracy.

Let us now take $\Gamma = \{-1, 0, 1\}^2 = \{(0,0), \pm(1,0), \pm(0,1), \pm(1,1), \pm(1,-1)\}$. Then, it is easy to see that transforming R and $\tilde{R}$ to 2-D using the prime coset sum with $p = 3$ produces two 2-D refinement masks $\mathcal{C}_{2,3}[R]$ and $\mathcal{C}_{2,3}[\tilde{R}]$ (cf. Figure 3.1) that are not only interpolatory with one accuracy, but also biorthogonal. ■

Example 3 (2-D Haar refinement mask with dilation 3). *Now let us consider the regular 1-D Haar refinement mask with dilation 3:*

$$\frac{1}{3}\left(1 + e^{-i\omega} + e^{-2i\omega}\right),$$

that is associated with the refinable function $\phi = \chi_{[0,1]}$, where $\chi_{[0,1]}$ is the characteristic function on $[0, 1]$. Let both R and $\tilde{R}$ be the above 1-D Haar refinement mask with dilation

3. *Then it is easy to see that R and $\tilde{R}$ are interpolatory and biorthogonal, and they have one accuracy.*

We use $\Gamma = \{0,1,2\}^2 = \{(0,0),(0,1),(0,2),(1,0),(1,1),(1,2),(2,0),(2,1),(2,2)\}$ *this time. By transforming R and $\tilde{R}$ to 2-D masks using the prime coset sum with $p = 3$, we see that $\mathcal{C}_{2,3}[R]$ and $\mathcal{C}_{2,3}[\tilde{R}]$ are still interpolatory and they still have one accuracy, but that they are no longer biorthogonal.* ■

3.3 Multi-D wavelet filter banks with fast algorithms

3.3.1 Theory

Suppose that S and U are 1-D biorthogonal refinement masks with dilation p, and that U is interpolatory. Since the n-D prime coset sum refinement masks $\mathcal{C}_{n,p}[S]$ and $\mathcal{C}_{n,p}[U]$ are not necessarily biorthogonal (cf. Example 3 in Section §3.2), it is not trivial to construct wavelet filter banks from $\mathcal{C}_{n,p}[S]$ and $\mathcal{C}_{n,p}[U]$ directly. We propose to use a recently developed method called effortless critical representation of Laplacian pyramid [44]. This method can construct wavelet filter banks from two refinement masks that are not necessarily biorthogonal, as long as one of them is interpolatory. Noting that $\mathcal{C}_{n,p}[U]$ is interpolatory (cf. Lemma 2), we apply this method to $\mathcal{C}_{n,p}[S]$ and $\mathcal{C}_{n,p}[U]$ to construct wavelet filter banks. As we will see later (cf. Section §3.3.2), similar to coset sum, the resulting wavelet filter banks using this method can be associated with fast algorithms, that are faster than the tensor product fast algorithms.

Since the method in [44] works for any dilation matrix Λ, below we present it for the general dilation matrix Λ with $q = |\det \Lambda|$. Let Γ and Γ^* be the complete set

of representatives of the distinct cosets of $\mathbb{Z}^n/\Lambda\mathbb{Z}^n$ and $2\pi(((\Lambda^*)^{-1}\mathbb{Z}^n)/\mathbb{Z}^n)$, respectively, containing 0. The following result is from [44] written in terms of our notation.

Result 3 *Suppose g and h are two n-D lowpass filters with dilation Λ, and h is interpolatory. Then the two n-D refinement masks defined as*

$$\tau(\omega) := \widehat{g}(\omega) + \Big(1 - \sum_{\gamma\in\Gamma^*} \widehat{g}(\omega+\gamma)\overline{\widehat{h}(\omega+\gamma)}\Big), \quad \tau^{\mathrm{d}}(\omega) := \widehat{h}(\omega),$$

for every $\omega \in \mathbb{T}^n$, and the n-D wavelet masks defined as

$$t_\nu(\omega) := e^{-i\omega\cdot\nu} - q\,\overline{(h(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)},$$

and

$$t^{\mathrm{d}}_\nu(\omega) := \frac{1}{q}\, e^{-i\omega\cdot\nu} - \overline{(g(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)}\,\widehat{h}(\omega),$$

for every $\omega \in \mathbb{T}^n$, and $\nu \in \Gamma' = \Gamma\backslash 0$, form the combined biorthogonal masks (cf. (3.3)). ■

Proof 4 *Result 3 is proved in [44], but under slightly different settings. For completeness, we provide an alternative proof that does not rely on the results of [44]. Our proof is placed in Appendix §3.5.4.* ■

Remark 5 *In fact, the results in [44] say that, if we assume that, in addition to the assumptions of Result 3, h has α_1 accuracy, g has α_2 accuracy, and α_3 flatness, then τ has at least $\min\{\alpha_1,\alpha_2,\alpha_3\}$ accuracy. In such a case, t_ν and t^{d}_ν, $\nu\in\Gamma'$, have at least $\min\{\alpha_1,\alpha_2,\alpha_3\}$ vanishing moments (cf. Section §3.1.2).* ■

For the rest of this section, we assume that the dilation is prime, i.e. $\Lambda = p\mathtt{I}_n$, and that the sets Γ and Γ^* are associated with the prime dilation, i.e., Γ and Γ^* are the complete

set of representatives of the distinct cosets of $\mathbb{Z}^n/p\mathbb{Z}^n$ and $2\pi((p^{-1}\mathbb{Z}^n)/\mathbb{Z}^n)$, respectively, containing 0. In particular, we have $q = |\det \Lambda| = p^n$ in this case.

Before presenting our main theorem, let us first define a map

$$\eta : F_p' \times \Gamma' \to \Gamma',$$

with $F_p' := F_p \backslash 0$, where F_p is a complete set of representatives of the distinct cosets of $\mathbb{Z}/p\mathbb{Z}$ that contains 0. For example, the set $\{0, 1, \cdots, p-1\}$ can be used for F_p. Let $(l, \nu) \in F_p' \times \Gamma' \subset \mathbb{Z} \times \mathbb{Z}^n$. Then there exists the unique multiplicative inverse $\rho(l) \in F_p'$ of l (cf. Remark 2 in Section §3.2). After computing the multiplication $\rho(l)\nu$ in the usual sense, we define $\eta(l, \nu)$ to be the element in $\Gamma' = \Gamma \backslash 0$ so that

$$\eta(l, \nu) \equiv \rho(l)\nu \,(\mathrm{mod}\, p\mathbb{Z}^n).$$

By the above conditions, $\eta(l, \nu)$ is uniquely well defined as an element in Γ' since $\rho(l)\nu$ is in $\mathbb{Z}^n$ but not in $p\mathbb{Z}^n$. For example, if $n = 2$, $p = 3$, $F_p = \{0, 1, 2\}$ and $\Gamma = \{0, 1, 2\}^2$, then $\eta(2, (1, 1)) = (2, 2)$ and $\eta(2, (2, 2)) = (1, 1)$.

Now we are ready to present our result.

Theorem 3 *Suppose that G and H are two 1-D lowpass filters with dilation p, and that H is interpolatory. Let $S := \widehat{G}$ and $U := \widehat{H}$ be the 1-D refinement masks associated with G and H, and let $\mathcal{C}_{n,p}$ be the prime coset sum. Define n-D biorthogonal refinement masks as*

$$\tau(\omega) := \mathcal{C}_{n,p}[S](\omega) + \left(1 - \sum_{\gamma \in \Gamma^*} \mathcal{C}_{n,p}[S](\omega+\gamma)\overline{\mathcal{C}_{n,p}[U](\omega+\gamma)}\right), \tau^{\mathrm{d}}(\omega) := \mathcal{C}_{n,p}[U](\omega),$$

for every $\omega \in \mathbb{T}^n$, and n-D wavelet masks as

$$t_\nu(\omega) := e^{-i\omega\cdot\nu}\left(1 - \frac{p}{p-1}\sum_{l \in F_p'} e^{i(\omega\cdot\eta(l,\nu))l}\,\overline{U_l\Big(p\omega \cdot \eta(l,\nu)\Big)}\right), \quad \nu \in \Gamma' \tag{3.8}$$

and

$$t_\nu^{\mathrm{d}}(\omega) := \frac{1}{p^n} e^{-i\omega\cdot\nu} \left(1 - \frac{p}{p-1} \sum_{l\in F_p'} e^{i(\omega\cdot\eta(l,\nu))l} \, \overline{S_l\Big(p\omega\cdot\eta(l,\nu)\Big)} \, \tau^{\mathrm{d}}(\omega) \right), \tag{3.9}$$

for $\nu \in \Gamma'$*, and for every* $\omega \in \mathbb{T}^n$*, where* $U_l(\xi) := (H(l{+}p\cdot))\hat{}\,(\xi)$*, and* $S_l(\xi) := (G(l{+}p\cdot))\hat{}\,(\xi)$*,* $\xi \in \mathbb{T}$.[3] *Then* $(\tau, (t_\nu)_{\nu\in\Gamma'})$ *and* $(\tau^{\mathrm{d}}, (t_\nu^{\mathrm{d}})_{\nu\in\Gamma'})$ *form n-D combined biorthogonal masks.* ■

Remark 6 *In the dyadic setting, i.e., when* $p = 2$*, one can take* $F_2 = \{0, 1\}$ *and* $\Gamma = \{0, 1\}^n$*. Then, since* 1 *is the only element in* F_2' *and* $\eta(1, \nu) = \nu$ *for all* $\nu \in \{0, 1\}^n\backslash 0$*, the n-D wavelet masks in (3.8) become*

$$\begin{aligned} t_\nu(\omega) &= e^{-i\omega\cdot\nu} - 2\,\overline{U_1\Big(2\omega\cdot\nu\Big)} \\ &= e^{-i\omega\cdot\nu} - 2\,\overline{e^{i\omega\cdot\nu}\Big(U(\omega\cdot\nu) - \frac{1}{2}\Big)} = e^{-i\omega\cdot\nu} - e^{-i\omega\cdot\nu}\Big(1 - 2\,\overline{U(\omega\cdot\nu+\pi)}\Big) \\ &= 2e^{-i\omega\cdot\nu}\,\overline{U(\omega\cdot\nu+\pi)}, \quad \nu \in \{0,1\}^n\backslash 0, \end{aligned}$$

where the second identity is from the definition of U_1 *and the third identity is from the fact that* U *is interpolatory. The above wavelet masks are the same as the wavelet masks in the coset sum wavelet system (cf. (3.4) in Result 2) up to a normalization factor. In fact, the exact forms of* t_ν^{d} *for coset sum wavelet system are also provided in [4], and similar calculation shows that they are the same as* t_ν^{d} *in (3.9) up to a normalization factor when* $p = 2$*. Hence we conclude that Theorem 3 reduces to the known result of the original coset sum case when* $p = 2$*.* ■

Remark 7 *We refer to the wavelet filter bank associated with the combined biorthogonal masks constructed in Theorem 3 as the prime coset sum wavelet filter bank. There are many*

[3] U_l and S_l can be interpreted as the polyphase decomposition of filter H and G, respectively (cf. Appendix §3.5.3).

potentially useful properties of the prime coset sum wavelet filter banks. One important property is that it can be implemented by fast algorithms (cf. Section §3.3.2). ■

Remark 8 *In addition to the assumptions of Theorem 3, if we assume that U has α_1 accuracy, S has α_2 accuracy, and α_3 flatness, then by Lemma 3 and Lemma 4, $\mathcal{C}_{n,p}[U]$ has at least α_1 accuracy, $\mathcal{C}_{n,p}[S]$ has at least $\min\{\alpha_2, \alpha_3\}$ accuracy, and at least α_3 flatness. Combining these with Remark 5, we conclude that τ has at least $\min\{\alpha_1, \alpha_2, \alpha_3\}$ accuracy, and t_ν and t_ν^{d}, $\nu \in \Gamma'$, have at least $\min\{\alpha_1, \alpha_2, \alpha_3\}$ vanishing moments.* ■

In order to prove Theorem 3, we use the following lemma which connects the *polyphase decomposition* of the 1-D lowpass filter H and the polyphase decomposition of the n-D prime coset sum lowpass filter h obtained from H. *Polyphase decomposition* is a useful tool in signal processing and we give a brief review in Appendix §3.5.3.

Lemma 5 *Let H be a 1-D lowpass filter with dilation p, and let h be the n-D lowpass filter obtained from H by applying the prime coset sum $\mathcal{C}_{n,p}$. Let the sets Γ' and F_p', and the map $\eta : \Gamma' \times F_p' \to \Gamma'$ be defined as before. Then for any $\nu \in \Gamma'$,*

$$(h(\nu + p\cdot))\widehat{\ }(p\omega) = \frac{1}{(p-1)p^{n-1}} \sum_{l \in F_p'} e^{i\omega\cdot(\nu - \eta(l,\nu)l)} \Big(H(l + p\cdot)\Big)\widehat{\ }(p\omega \cdot \eta(l,\nu)), \omega \in \mathbb{T}^n. \quad ■$$

Proof 5 *First it is easy to see that (cf. (3.15) in Appendix §3.5.3)*

$$\widehat{H}(\omega) \;=\; \sum_{l \in F_p} e^{-i\omega l} \Big(H(l + p\cdot)\Big)\widehat{\ }(p\omega), \quad \omega \in \mathbb{T}.$$

Using this identity and the definition of prime coset sum, we get

$$\begin{aligned} \widehat{h}(\omega) &= \frac{1}{(p-1)p^{n-1}} \left(1 - p^{n-1} + \sum_{\nu \in \Gamma'} \widehat{H}(\omega \cdot \nu)\right), \quad \omega \in \mathbb{T}^n \\ &= \frac{1}{(p-1)p^{n-1}} \left(1 - p^{n-1} + \sum_{\nu \in \Gamma'} \sum_{l \in F_p} e^{-i\omega\cdot\nu l} \Big(H(l + p\cdot)\Big)\widehat{\ }(p\omega \cdot \nu)\right). \end{aligned} \tag{3.10}$$

Next we use another identity that can be quickly derived (cf. (48) in [44]):

$$(h(\nu + p\cdot))\widehat{\ }(p\omega) = \frac{1}{p^n}\sum_{\gamma\in\Gamma^*} e^{i(\omega+\gamma)\cdot\nu}\,\widehat{h}(\omega+\gamma), \quad \omega\in\mathbb{T}^n. \tag{3.11}$$

By using (3.10), (3.11), and the fact that $\Big(H(l+p\cdot)\Big)\widehat{\ }(p(\omega+\gamma)\cdot\tilde{\nu}) = \Big(H(l+p\cdot)\Big)\widehat{\ }(p\omega\cdot\tilde{\nu})$*, for any* $l\in F_p$*,* $\omega\in\mathbb{T}^n$*,* $\gamma\in\Gamma^*$ *and* $\tilde{\nu}\in\Gamma'$*, we obtain* $(h(\nu+p\cdot))\widehat{\ }(p\omega) =$

$$\frac{1}{p^n}\sum_{\gamma\in\Gamma^*} e^{i(\omega+\gamma)\cdot\nu}\frac{1}{(p-1)p^{n-1}}\left(1-p^{n-1}+\sum_{\tilde{\nu}\in\Gamma'}\sum_{l\in F_p} e^{-i(\omega+\gamma)\cdot\tilde{\nu}l}\Big(H(l+p\cdot)\Big)\widehat{\ }(p\omega\cdot\tilde{\nu})\right).$$

Then we use the following simple identity (cf. (3.20)):

$$\sum_{\gamma\in\Gamma^*} e^{i\gamma\cdot\nu} = p^n\delta_{\nu,0} = \begin{cases} p^n, & \text{if } \nu = 0,\\ 0, & \text{if } \nu\in\Gamma'\backslash 0,\end{cases}$$

to get

$$\begin{aligned}
&(h(\nu+p\cdot))\widehat{\ }(p\omega)\\
=&\frac{1}{p^n}\sum_{\gamma\in\Gamma^*} e^{i(\omega+\gamma)\cdot\nu}\frac{1}{(p-1)p^{n-1}}\sum_{\tilde{\nu}\in\Gamma'}\sum_{l\in F_p'} e^{-i(\omega+\gamma)\cdot\tilde{\nu}l}\Big(H(l+p\cdot)\Big)\widehat{\ }(p\omega\cdot\tilde{\nu})\\
=&\frac{1}{p^n}\frac{1}{(p-1)p^{n-1}}\sum_{\tilde{\nu}\in\Gamma'}\sum_{l\in F_p'} e^{i\omega\cdot(\nu-\tilde{\nu}l)}\Big(H(l+p\cdot)\Big)\widehat{\ }(p\omega\cdot\tilde{\nu})\sum_{\gamma\in\Gamma^*} e^{i\gamma\cdot(\nu-\tilde{\nu}l)}, \quad \omega\in\mathbb{T}^n.
\end{aligned}$$

Noting that $\sum_{\gamma\in\Gamma^*} e^{i\gamma\cdot(\nu-\tilde{\nu}l)} = p^n$ *if* $\tilde{\nu} = \eta(l,\nu)$*, and it is equal to* 0 *otherwise, we obtain*

$$(h(\nu+p\cdot))\widehat{\ }(p\omega) = \frac{1}{(p-1)p^{n-1}}\sum_{l\in F_p'} e^{i\omega\cdot(\nu-\eta(l,\nu)l)}\Big(H(l+p\cdot)\Big)\widehat{\ }(p\omega\cdot\eta(l,\nu)), \omega\in\mathbb{T}^n,$$

as desired. ■

We now present the proof of Theorem 3.

Proof 6 (Proof of Theorem 3) *Let* g *and* h *be the* n*-D lowpass filters associated with refinement masks* $\mathcal{C}_{n,p}[S]$ *and* $\mathcal{C}_{n,p}[U]$*. Since* U *is interpolatory, by Lemma 2,* $\mathcal{C}_{n,p}[U]$ *is also*

interpolatory, i.e., h is interpolatory. Therefore, we can obtain the combined biorthogonal masks by using Result 3. By setting $\widehat{g} := \mathcal{C}_{n,p}[S]$ and $\widehat{h} := \mathcal{C}_{n,p}[U]$ in Result 3, we obtain that, for every $\omega \in \mathbb{T}^n$,

$$\begin{aligned}\tau(\omega) &= \widehat{g}(\omega) + \Big(1 - \sum_{\gamma\in\Gamma^*} \widehat{g}(\omega+\gamma)\overline{\widehat{h}(\omega+\gamma)}\Big) \\ &= \mathcal{C}_{n,p}[S](\omega) + \left(1 - \sum_{\gamma\in\Gamma^*} \mathcal{C}_{n,p}[S](\omega+\gamma)\overline{\mathcal{C}_{n,p}[U](\omega+\gamma)}\right),\end{aligned}$$

and

$$\tau^{\mathrm{d}}(\omega) = \widehat{h}(\omega) = \mathcal{C}_{n,p}[U](\omega).$$

Since, in this case, $\Lambda = p\mathtt{I}_n$ and $q = p^n$, the n-D wavelet masks t_ν, $\nu \in \Gamma'$, are

$$\begin{aligned}t_\nu(\omega) &= e^{-i\omega\cdot\nu} - q\,\overline{(h(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)} \\ &= e^{-i\omega\cdot\nu} - p^n\,\overline{(h(\nu+p\cdot))\widehat{\ }(p\omega)}, \quad \omega \in \mathbb{T}^n.\end{aligned}$$

Since H is the 1-D filter associated with U and h is the n-D filter associated with $\mathcal{C}_{n,p}[U]$, by Lemma 5, we have

$$(h(\nu+p\cdot))\widehat{\ }(p\omega) = \frac{1}{(p-1)p^{n-1}} \sum_{l\in F_p'} e^{i\omega\cdot(\nu-\eta(l,\nu)l)} \Big(H(l+p\cdot)\Big)\widehat{\ }(p\omega\cdot\eta(l,\nu)).$$

Therefore,

$$\begin{aligned}t_\nu(\omega) &= e^{-i\omega\cdot\nu} - p^n\,\overline{\frac{1}{(p-1)p^{n-1}} \sum_{l\in F_p'} e^{i\omega\cdot(\nu-\eta(l,\nu)l)} \Big(H(l+p\cdot)\Big)\widehat{\ }(p\omega\cdot\eta(l,\nu))} \\ &= e^{-i\omega\cdot\nu} - \frac{p}{p-1} \sum_{l\in F_p'} e^{i\omega\cdot(\eta(l,\nu)l-\nu)}\,\overline{\Big(H(l+p\cdot)\Big)\widehat{\ }(p\omega\cdot\eta(l,\nu))} \\ &= e^{-i\omega\cdot\nu}\left(1 - \frac{p}{p-1} \sum_{l\in F_p'} e^{i\omega\cdot\eta(l,\nu)l}\,\overline{U_l\Big(p\omega\cdot\eta(l,\nu)\Big)}\right), \quad \omega \in \mathbb{T}^n.\end{aligned}$$

The wavelet masks t^{d}_ν, $\nu \in \Gamma'$, in (3.9) can be obtained by applying similar arguments to the general from of t^{d}_ν, $\nu \in \Gamma'$, in Result 3. This concludes that $(\tau, (t_\nu)_{\nu\in\Gamma'})$ and $(\tau^{\mathrm{d}}, (t^{\mathrm{d}}_\nu)_{\nu\in\Gamma'})$ defined as in Theorem 3 form n-D combined biorthogonal masks. ■

The following corollary of Theorem 3 may be useful on its own in some contexts.

Corollary 1 *Suppose that S and U are two 1-D refinement masks with dilation p, and that U is interpolatory. Let $\mathcal{C}_{n,p}$ be the prime coset sum. Then the two n-D refinement masks $\mathcal{C}_{n,p}[U]$ and*

$$\mathcal{C}_{n,p}[S] + \left(1 - \sum_{\gamma\in\Gamma^*} \mathcal{C}_{n,p}[S](\cdot+\gamma)\overline{\mathcal{C}_{n,p}[U](\cdot+\gamma)}\right)$$

with dilation $p\mathtt{I}_n$ are biorthogonal. ■

Remark 9 *Of the two prime coset sum refinement masks $\mathcal{C}_{n,p}[S]$ and $\mathcal{C}_{n,p}[U]$, only the non-interpolatory mask $\mathcal{C}_{n,p}[S]$ is modified by adding $1 - \sum_{\gamma\in\Gamma^*} \mathcal{C}_{n,p}[S](\cdot+\gamma)\overline{\mathcal{C}_{n,p}[U](\cdot+\gamma)}$. We note that the statement of Corollary 1 holds true trivially for the case when $\mathcal{C}_{n,p}[S]$ and $\mathcal{C}_{n,p}[U]$ are already biorthogonal, since $1 - \sum_{\gamma\in\Gamma^*} \mathcal{C}_{n,p}[S](\cdot+\gamma)\overline{\mathcal{C}_{n,p}[U](\cdot+\gamma)} = 0$ in such a case. One such case is when S and U are biorthogonal and $p = 2$ (cf. Result 1(b)).* ■

Next we illustrate our findings in two examples.

Example 4 (Centered n-D Haar combined biorthogonal masks with dilation 3). *Let us consider the centered 1-D Haar refinement mask with dilation 3 as in Example 1. We let both S and U be*

$$S(\omega) = U(\omega) := \frac{1}{3}\left(e^{i\omega} + 1 + e^{-i\omega}\right).$$

Then they are both interpolatory with one accuracy. Now let us take $\Gamma = \{-1, 0, 1\}^n$ *and* $\Gamma^* = \frac{2\pi}{3}\{-1, 0, 1\}^n$ *for any dimension* n*. Then by Theorem 3 the* n*-D biorthogonal refinement masks*

$$\tau(\omega) = \tau^{\mathrm{d}}(\omega) = \frac{1}{3^n} \sum_{\nu \in \Gamma} e^{-i\omega\cdot\nu}, \quad \omega \in \mathbb{T}^n,$$

and n*-D wavelet masks*

$$t_\nu(\omega) = e^{-i\omega\cdot\nu} - 1, \quad t_\nu^{\mathrm{d}}(\omega) = \frac{1}{3^n}\, e^{-i\omega\cdot\nu} - \frac{1}{9^n} \sum_{\mu \in \Gamma} e^{-i\omega\cdot\mu}, \quad \omega \in \mathbb{T}^n,$$

for $\nu \in \Gamma'$*, form* n*-D combined biorthogonal masks. These combined biorthogonal masks are studied also in [76]. By direct computation, we see that both* τ *and* τ^{d} *have one accuracy, and that both* t_ν *and* t_ν^{d} *have one vanishing moment for any* $\nu \in \Gamma'$*. The number of nonzero entries, or the support of the filter associated with* t_ν *is only* 2 *for any* $\nu \in \Gamma'$ *and any dimension* n*.* ■

Example 5 (2-D combined biorthogonal masks with higher vanishing moments). *Let* U *be a* 1*-D interpolatory refinement mask with dilation* 3 *and accuracy* 4 [4]

$$U(\omega) := \frac{1}{3}\left(-\frac{4}{81}e^{5i\omega} - \frac{5}{81}e^{4i\omega} + \frac{30}{81}e^{2i\omega} + \frac{60}{81}e^{i\omega} + 1 + \frac{60}{81}e^{-i\omega} + \frac{30}{81}e^{-2i\omega} - \frac{5}{81}e^{-4i\omega} - \frac{4}{81}e^{-5i\omega}\right).$$

Let S *be defined as in Example 4. We take* $\Gamma = \{-1, 0, 1\}^2$ *and* $\Gamma^* = \frac{2\pi}{3}\{-1, 0, 1\}^2$*. Then by Theorem 3 the* 2*-D biorthogonal refinement masks*

$$\tau(\omega) = \frac{1}{9}\left(\frac{83}{27} + \sum_{\nu \in \Gamma'} e^{-i\omega\cdot\nu} - \frac{25}{81}\sum_{\nu \in \Gamma'} e^{-3i\omega\cdot\nu} + \frac{4}{81}\sum_{\nu \in \Gamma'} e^{-6i\omega\cdot\nu}\right), \quad \omega \in \mathbb{T}^2,$$

$$\tau^{\mathrm{d}}(\omega) = \frac{1}{9}\left(1 + \frac{60}{81}\sum_{\nu \in \Gamma'} e^{-i\omega\cdot\nu} + \frac{30}{81}\sum_{\nu \in \Gamma'} e^{-2i\omega\cdot\nu} - \frac{5}{81}\sum_{\nu \in \Gamma'} e^{-4i\omega\cdot\nu} - \frac{4}{81}\sum_{\nu \in \Gamma'} e^{-5i\omega\cdot\nu}\right), \quad \omega \in \mathbb{T}^2,$$

and 2*-D wavelet masks*

$$t_\nu(\omega) = e^{-i\omega\cdot\nu} + \frac{5}{81}e^{3i\omega\cdot\nu} - \frac{60}{81} - \frac{30}{81}e^{-3i\omega\cdot\nu} + \frac{4}{81}e^{-6i\omega\cdot\nu}, \quad \omega \in \mathbb{T}^2,$$

[4] U is obtained from Example 1) in Section V. A. of [49].

$$\begin{array}{cccccccccccc}
-\frac{4}{81} & -\frac{5}{81} & 0 & \frac{30}{81} & \frac{60}{81} & \mathbf{1} & \frac{60}{81} & \frac{30}{81} & 0 & -\frac{5}{81} & -\frac{4}{81} \\
\multicolumn{11}{c}{\text{Filter associated with } U} \\
\multicolumn{11}{c}{\downarrow \mathcal{C}_{2,3} \text{ (Prime coset sum)}} \\
-\frac{4}{81} & 0 & 0 & 0 & 0 & -\frac{4}{81} & 0 & 0 & 0 & 0 & -\frac{4}{81} \\
0 & -\frac{5}{81} & 0 & 0 & 0 & -\frac{5}{81} & 0 & 0 & 0 & -\frac{5}{81} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{30}{81} & 0 & \frac{30}{81} & 0 & \frac{30}{81} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{60}{81} & \frac{60}{81} & \frac{60}{81} & 0 & 0 & 0 & 0 \\
-\frac{4}{81} & -\frac{5}{81} & 0 & \frac{30}{81} & \frac{60}{81} & \mathbf{1} & \frac{60}{81} & \frac{30}{81} & 0 & -\frac{5}{81} & -\frac{4}{81} \\
0 & 0 & 0 & 0 & \frac{60}{81} & \frac{60}{81} & \frac{60}{81} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{30}{81} & 0 & \frac{30}{81} & 0 & \frac{30}{81} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{5}{81} & 0 & 0 & 0 & -\frac{5}{81} & 0 & 0 & 0 & -\frac{5}{81} & 0 \\
-\frac{4}{81} & 0 & 0 & 0 & 0 & -\frac{4}{81} & 0 & 0 & 0 & 0 & -\frac{4}{81} \\
\multicolumn{11}{c}{\text{Filter associated with } \tau^{\mathrm{d}} = \mathcal{C}_{2,3}[U]}
\end{array}$$

Figure 3.2: Lowpass filters associated with the masks U and τ^{d} in Example 5.

$$t^{\mathrm{d}}_{\nu}(\omega) = \frac{1}{9}\left(e^{-i\omega\cdot\nu} - \tau^{\mathrm{d}}(\omega)\right), \quad \omega \in \mathbb{T}^2,$$

for $\nu \in \Gamma'$, form 2-D combined biorthogonal masks (cf. Figure 3.2 for the filters associated with U and τ^{d}). Direct computation shows that τ has one accuracy, τ^{d} has 4 accuracy, t_ν, $\nu \in \Gamma'$, have 4 vanishing moments, and t^{d}_ν, $\nu \in \Gamma'$, have one vanishing moment. The support of the filter associated with t_ν is only 5 for any $\nu \in \Gamma'$. ■

3.3.2 Algorithms

Theorem 3 provides only one of many ways to obtain the non-redundant wavelet filter bank, given the two n-D refinement masks $\mathcal{C}_{n,p}[S]$ and $\mathcal{C}_{n,p}[U]$. However, the resulting prime coset sum wavelet filter bank can be associated with fast algorithms that are faster than the usual tensor product ones. Below we present these fast prime coset sum algorithms.

Fast Prime Coset Sum Wavelet Algorithms. Let G and H be two 1-D lowpass filters with dilation p, where H is interpolatory. In presenting our algorithms, we use the set F_p and the map η that we defined in Section §3.3.1.

input $y_J : \mathbb{Z}^n \to \mathbb{R}$

(1) Decomposition Algorithm: computing y_{j-1}, $w_{\nu,j-1}$, $\nu \in \Gamma'$ from y_j

`for` $j = J, J-1, \ldots, 1$

`for` $\nu \in \Gamma'$ `and` $k \in \mathbb{Z}^n$

$$w_{\nu,j-1}(k) = y_j(pk+\nu) - \frac{1}{p-1} \sum_{l \in F_p'} \sum_{m \equiv l} H(m) y_j(pk + \nu - \eta(l,\nu)m) \qquad \texttt{(i)}$$

`end`

`for` $k \in \mathbb{Z}^n$

$$y_{j-1}(k) = y_j(pk) + \frac{1}{(p-1)p^n} \sum_{\nu \in \Gamma'} \sum_{l \in F_p'} \sum_{m \equiv l} G(m) w_{\nu,j-1}(k - \frac{\nu - \eta(l,\nu)m}{p}) \qquad \texttt{(ii)}$$

`end`

`end`

(2) Reconstruction Algorithm: computing y_j from y_{j-1}, $w_{\nu,j-1}$, $\nu \in \Gamma'$

`for` $j = 1, \ldots, J-1, J$

`for` $k \in \mathbb{Z}^n$

$$y_j(pk) = y_{j-1}(k) - \frac{1}{(p-1)p^n} \sum_{\nu\in\Gamma'} \sum_{l\in F_p'} \sum_{m\equiv l} G(m) w_{\nu,j-1}(k - \frac{\nu - \eta(l,\nu)m}{p}) \quad \texttt{(iii)}$$

`end`

`for` $\nu \in \Gamma'$ `and` $k \in \mathbb{Z}^n$

$$y_j(pk+\nu) = w_{\nu,j-1}(k) + \frac{1}{p-1} \sum_{l\in F_p'} \sum_{m\equiv l} H(m) y_j(pk + \nu - \eta(l,\nu)m) \quad \texttt{(iv)}$$

`end`

`end`

For decomposition, we compute the coarse coefficients y_{j-1} and wavelet coefficients $w_{\nu,j-1}$, $\nu \in \Gamma'$, from y_j. To obtain $w_{\nu,j-1}$, $\nu \in \Gamma'$, we apply the filter associated with t_ν, $\nu \in \Gamma'$ to y_j, followed by downsampling with respect to the dilation matrix $\Lambda = p\mathtt{I}_n$, as is typically done in wavelet decomposition. Since t_ν, $\nu \in \Gamma'$, are written in terms of U_l, $l \in F_p'$, and since U_l can be written in terms of 1-D filter H, we obtain the formula for Step `(i)`. The proof of the identity in Step (i) is given in Appendix §3.5.5, in which the concept of polyphase decomposition (cf. Appendix §3.5.3) is used.

A key step of our decomposition algorithm is Step `(ii)`. Typically, to obtain y_{j-1}, one needs to apply the filter associated with τ to y_j, followed by downsampling. However, since here $\tau = \mathcal{C}_{n,p}[S] + \left(1 - \sum_{\gamma\in\Gamma^*} \mathcal{C}_{n,p}[S](\cdot+\gamma)\overline{\mathcal{C}_{n,p}[U](\cdot+\gamma)}\right)$ (cf. Theorem 3), contrary to the filter associated with the first part of τ, i.e. $\mathcal{C}_{n,p}[S]$, it is not clear how the filter associated with the rest of the mask τ, i.e. $1 - \sum_{\gamma\in\Gamma^*} \mathcal{C}_{n,p}[S](\cdot+\gamma)\overline{\mathcal{C}_{n,p}[U](\cdot+\gamma)}$, would look like. As a result, the support of the filter associated with τ could be large. Therefore, the algorithm may not be faster than other wavelet algorithms if we use the filter associated with τ directly. However, by using the polyphase representation (cf. Appendix §3.5.3), one can show that y_{j-1} can also be derived by applying the filter associated with $\mathcal{C}_{n,p}[S]$ (the

first part of τ) to $w_{\nu,j-1}$, $\nu \in \Gamma'$. This is our Step `(ii)`, and the details of exactly how it is done are written in Appendix §3.5.5.

Our reconstruction algorithm is not the same as the typical wavelet reconstruction procedure either. We recall that the typical wavelet reconstruction is conducted by applying the reconstruction filters to y_{j-1} and $w_{\nu,j-1}$, $\nu \in \Gamma'$, upsampling them, and then summing them up. We reconstruct the signal by simply reversing Step `(i)` and `(ii)`. Step `(iii)` is a reverse procedure of Step `(ii)` that can always be performed. Step `(iv)` is a reverse procedure of Step `(i)`, and it is possible because the only y_j needed in the right-hand side of Step `(iv)` is $y_j(pk)$, which is already computed in Step `(iii)`.

Complexity. Next we discuss the complexity of the fast prime coset sum wavelet algorithms. We measure the complexity by counting the number of multiplicative operations needed in a complete cycle of 1-level-down decomposition and 1-level-up reconstruction, meaning the number of operations needed to fully derive y_{j-1} and $w_{\nu,j-1}$, $\nu \in \Gamma'$ from y_j, and to get back y_j. Here we only compute the number of multiplicative operations such as multiplication and division, as computing additive operations gives a similar result.

Suppose that at level j, we have input data y_j with N data points. For simplicity, we assume that N is a multiple of p^n, where p is the dilation and n is the spatial dimension. Then after 1-level-down decomposition, we obtain N/p^n coarse coefficients y_{j-1} in Step `(ii)`, and N/p^n wavelet coefficients $w_{\nu,j-1}$ for each $\nu \in \Gamma'$ in Step `(i)`. We reconstruct the input data y_j from coarse coefficients y_{j-1} and wavelet coefficients $w_{\nu,j-1}$, $\nu \in \Gamma'$. In particular, we obtain N/p^n original data $y_j(pk)$ in Step `(iii)` and N/p^n original data $y_j(pk+\nu)$ for each $\nu \in \Gamma'$ in Step `(iv)`.

Suppose α and β are the number of nonzero entries in the 1-D lowpass filter G and H, respectively. Recall that H is interpolatory. Let

$$\tilde{\alpha} := \#\{G(m) : \ G(m) \neq 0 \text{ and } m \equiv l \,(\mathrm{mod}\, p\mathbb{Z}) \text{ for some } l \in F_p'\}.$$

Given the N data points of the input data y_j, the number of multiplicative operations needed in a complete cycle of 1-level-down decomposition and 1-level-up reconstruction is the sum of

- $2\beta(p^n-1)\frac{N}{p^n}$ [for Step `(i)` and `(iv)`], and
- $2\Big((p^n-1)\tilde{\alpha}+n+1\Big)\frac{N}{p^n}$ [for Step `(ii)` and `(iii)`].

Therefore, as a result, the complexity of the fast prime coset sum wavelet algorithms is

$$\left(\frac{2(p^n-1)\beta+2(p^n-1)\tilde{\alpha}+2n+2}{p^n}\right)N. \tag{3.12}$$

Since $\tilde{\alpha} \leq \frac{p-1}{p}(\alpha+1)$, this complexity is bounded above by

$$\Big(2\beta+2\,\frac{p-1}{p}(\alpha+1)+1\Big)N. \qquad \blacksquare$$

Recall that in dyadic case, the fast tensor product wavelet algorithms have complexity $(\alpha+\beta)nN$, where α and β are the number of nonzero entries of 1-D lowpass filters, n is the spatial dimension and N is the data size (see, for example, [4]). Therefore, the algorithm has linear complexity, i.e., $\sim CN$, with the data size N, where C is some constant that does not depend on N. We refer to this constant as the *complexity constant*. The complexity constant for fast tensor product wavelet algorithm is $C_{TP} = (\alpha+\beta)n$. In particular, it grows linearly with the dimension n. Now let us consider the fast prime coset

sum wavelet algorithm. In dyadic case, i.e., when $p = 2$, the complexity is bounded above by $(\alpha + 2\beta + 2)N$. Therefore, the complexity constant is $C_{PCS} = \alpha + 2\beta + 2$, which does not increase as dimension n increases. Furthermore, since $\alpha \geq 2$, we have $C_{PCS} \leq C_{TP}$ for all $n \geq 2$, which suggests that our fast prime coset sum algorithms can be much faster, at least in theory, than the fast tensor product algorithms when n is large.

Our fast algorithms with $p = 2$ are different from the original fast coset sum algorithms in [4], which results in a different complexity constant for the coset sum case. The complexity constant for the fast coset sum algorithms is $C_{CS} = \frac{3}{2}\alpha + 2\beta$, and as a result, we have $C_{PCS} \leq C_{CS}$ as long as $\alpha \geq 4$.

There are a couple of factors that contribute to make our algorithms this fast. Firstly, the number of nonzero entries in the n-D filter associated with t_ν, $\nu \in \Gamma'$, is essentially the same as that of the 1-D filter H (cf. Step (`i`)). Secondly, our decomposition algorithm is performed by bypassing the filter associated with τ (cf. Step (`ii`)), which could have large support, in general. Finally, the reconstruction algorithm has trivial reconstruction steps, which completely bypass the filters associated with t^{d}_ν, $\nu \in \Gamma'$ (cf. Step (`iii`) and (`iv`)).

We now illustrate our findings using some examples.

Example 6 (Fast prime coset sum wavelet algorithms for centered n-D Haar with dilation 3). *Let us consider the centered n-D Haar combined biorthogonal masks with dilation 3 constructed in Example 4. In this case, the 1-D filter G and H are given as*

$$G(K) = H(K) = \begin{cases} 1, & \text{if } K = 0, \\ 1, & \text{if } K = \pm 1, \\ 0, & \text{otherwise.} \end{cases}$$

Then one can follow Step (i) – (iv) *with this pair of* G *and* H *to perform the fast algorithms. In this case,* $\alpha = \beta = 3$, $\tilde{\alpha} = 2$ *and* $p = 3$. *Hence for any dimension* n, *and input data of size* N, *the algorithms have complexity*

$$\left(\frac{6(3^n-1)+4(3^n-1)+2n+2}{3^n}\right) N \leq 11N.$$

Hence the complexity constant in this case is 11, *and it does not grow as the dimension* n *grows.* ■

Example 7 (Fast prime coset sum wavelet algorithms for 2-D wavelets with higher vanishing moments). *Let us consider the 2-D combined biorthogonal masks constructed in Example 5. In this case, the 1-D filter* G *and* H *are given as*

$$G(K) = \begin{cases} 1, & \text{if } K = 0, \\ 1, & \text{if } K = \pm 1, \\ 0, & \text{otherwise}, \end{cases} \quad H(K) = \begin{cases} 1, & \text{if } K = 0, \\ \frac{60}{81}, & \text{if } K = \pm 1, \\ \frac{30}{81}, & \text{if } K = \pm 2, \\ -\frac{5}{81}, & \text{if } K = \pm 4, \\ -\frac{4}{81}, & \text{if } K = \pm 5, \\ 0, & \text{otherwise.} \end{cases}$$

Then this pair of G *and* H *can be used in Step* (i) – (iv) *to implement the fast algorithms for the wavelet filter bank constructed in Example 5. In particular, since* $\alpha = 3$, $\beta = 9$, $\tilde{\alpha} = 2$, $p = 3$ *and* $n = 2$, *the fast algorithms have complexity*

$$\left(\frac{18(3^2-1)+4(3^2-1)+6}{3^2}\right) N \leq 21N,$$

for any input data of size N. *Hence the complexity constant in this case is* 21. ■

3.4 Conclusion

In this paper we introduced a method called prime coset sum to construct multi-D refinement masks from 1-D refinement masks. This method is a generalization of the existing method, the coset sum ([4]), that works only for the dyadic dilations. We showed that for a prime dilation, the prime coset sum method maintains many important properties from the 1-D refinement masks, such as interpolatory property, and under some conditions, the accuracy number. More importantly, the prime coset sum refinement masks can be used to construct wavelet filer banks with fast algorithms. Similar to the coset sum method for dyadic case, the prime coset sum fast algorithms have complexity constant that does not increase as the spatial dimension n increases. This is contrary to the tensor product method, since its complexity constant increases linearly with the spatial dimension.

3.5 Appendix

3.5.1 Proof of Lemma 2 in section §3.2

Suppose H and h are the filters associated with masks R and $\mathcal{C}_{n,p}[R]$. If R is interpolatory, by (3.2), $H(0) = 1$, and $H(K) = 0$ for any $K \in p\mathbb{Z}\backslash 0$. Then, by (3.7), $h(0) = \frac{1}{p-1}(p - p^n + (p^n - 1)H(0)) = 1$, and $h(k) = \frac{1}{p-1}\sum_{l\in W_k} H(l)$ for any $k \neq 0$. Since for each $k \in p\mathbb{Z}^n\backslash 0$, every element l in the set $W_k = \{l \in \mathbb{Z}\backslash 0 : k = l\nu \text{ for some } \nu \in \Gamma'\}$ must lie in $p\mathbb{Z}\backslash 0$, we see that $h(k) = \frac{1}{p-1}\sum_{l\in W_k} H(l) = 0$ for any $k \in p\mathbb{Z}^n\backslash 0$. Hence $\mathcal{C}_{n,p}[R]$ is interpolatory.

3.5.2 Proof of Lemma 3 in section §3.2

First we note that $\mathcal{C}_{n,p}[R]$ has at least accuracy number one, since R has at least accuracy number one and $\mathcal{C}_{n,p}$ is defined so that it preserves positive accuracy.

Let F_p^* be a complete set of representatives of the distinct cosets of $2\pi((p^{-1}\mathbb{Z})/\mathbb{Z})$ containing 0. Since the order of zeros of R at $\xi \in F_p^* \backslash 0$ is m_1, and the order of zeros of $1-R$ at the origin is m_2, we have, for any integer $1 \le k \le \min\{m_1, m_2\} - 1$,

$$(D^k R)(\xi) = 0, \quad \text{for any } \xi \in F_p^*. \tag{3.13}$$

Thus, for any $\gamma \in \Gamma^*$ and any $\mu \in \mathbb{N}^n$ with $1 \le |\mu| \le \min\{m_1, m_2\} - 1$, where $|\mu| := \mu_1 + \cdots + \mu_n$, we get

$$\begin{aligned}(D^\mu \mathcal{C}_{n,p}[R])(\gamma) &= \frac{1}{(p-1)p^{n-1}} \sum_{\nu \in \Gamma'} (D^\mu [R(\omega \cdot \nu)]) \mid_{\omega=\gamma} \\ &= \frac{1}{(p-1)p^{n-1}} \sum_{\nu \in \Gamma'} \left(\prod_{j=1}^{n} \nu_j^{\mu_j} \right) (D^{|\mu|} R)(\gamma \cdot \nu) = 0,\end{aligned}$$

where the last equality is from (3.13) and the fact that $\gamma \cdot \nu \,(\mathrm{mod}\, p\mathbb{Z})$ belongs to F_p^*. This implies the accuracy number of $\mathcal{C}_{n,p}[R]$ is at least $\min\{m_1, m_2\}$.

3.5.3 Review of polyphase representation of wavelet filter banks

The polyphase decomposition in [77] is widely used in signal processing. We briefly review some relevant concepts in polyphase decomposition in terms of our notation and terminology, and refer other papers (e.g. [44, 78]) for details.

As before, we use Λ to denote the dilation matrix, and q to denote $|\det \Lambda|$. The polyphase decomposition transforms a filter (or signal) into q filters (or signals) running at the sampling rate $1/q$. Let Γ be a complete set of representatives of the distinct cosets of

$\mathbb{Z}^n/\Lambda\mathbb{Z}^n$ containing 0, and let $\Gamma' = \Gamma\backslash 0$. For example, for the scalar dilation with λ, the set $\{0, 1, \cdots, \lambda - 1\}^n$ can be used for Γ.

The *polyphase decomposition of a synthesis filter* h is defined as the Fourier series of $h(\nu + \Lambda\cdot)$, $\nu \in \Gamma$:

$$\mathtt{H}_\nu(\omega) := (h(\nu + \Lambda\cdot))\widehat{\ }(\omega) = \frac{1}{q}\sum_{k\in\mathbb{Z}^n} h(\nu + \Lambda k)e^{-ik\cdot\omega}, \quad \omega \in \mathbb{T}^n, \tag{3.14}$$

and the *polyphase representation of a synthesis filter* h is defined as the column q-vector of the form

$$\mathtt{H}(\omega) := [\mathtt{H}_{\nu_0}(\omega), \mathtt{H}_{\nu_1}(\omega), \cdots, \mathtt{H}_{\nu_{q-1}}(\omega)]^T, \quad \omega \in \mathbb{T}^n,$$

where $\nu_0 = 0$ and ν_j, $j = 1, \ldots, q-1$, are the ordered elements of the set Γ'. Then it is easy to see that the Fourier series of h can be written in terms of the polyphase decomposition of h as follows:

$$\widehat{h}(\omega) = \sum_{\nu\in\Gamma} e^{-i\omega\cdot\nu}\mathtt{H}_\nu(\Lambda^*\omega). \tag{3.15}$$

Similarly, the *polyphase decomposition of an analysis filter* g is defined as the complex conjugate of the Fourier series of $g(\nu + \Lambda\cdot)$, $\nu \in \Gamma$:

$$\mathtt{G}_\nu(\omega) := \overline{(g(\nu + \Lambda\cdot))\widehat{\ }(\omega)} = \frac{1}{q}\sum_{k\in\mathbb{Z}^n} g(\nu - \Lambda k)e^{-ik\cdot\omega}, \quad \omega \in \mathbb{T}^n, \tag{3.16}$$

and the *polyphase representation of an analysis filter* g is defined as the row q-vector of the form

$$\mathtt{G}(\omega) := [\mathtt{G}_{\nu_0}(\omega), \mathtt{G}_{\nu_1}(\omega), \cdots, \mathtt{G}_{\nu_{q-1}}(\omega)], \quad \omega \in \mathbb{T}^n,$$

and, as a result, we have the identity

$$\overline{\widehat{g}(\omega)} = \sum_{\nu\in\Gamma} e^{i\omega\cdot\nu}\mathtt{G}_\nu(\Lambda^*\omega).$$

Under these notations, it is easy to see that h and g are biorthogonal if and only if $\mathtt{G}(\omega)\mathtt{H}(\omega) = 1/q$.

A filter bank (that is non-redunant with perfect reconstruction property) can be represented by two $q \times q$ polyphase matrices $\mathtt{A}(\omega)$ and $\mathtt{S}(\omega)$ that satisfy $\mathtt{S}(\omega)\mathtt{A}(\omega) = (1/q)\mathtt{I}_q$. The row vectors of $\mathtt{A}(\omega)$ represent the polyphase representation of analysis filters, where the first row corresponding to the lowpass filter and the rest to the highpass filters. The column vectors of $\mathtt{S}(\omega)$ represent the polyphase representation of synthesis filters, where the first column corresponding to the lowpass filter and the rest to the highpass filters.

We finish this subsection by stating Result 3 in terms of the polyphase representation, as it will be useful in the later part of the paper.

Result 4 (Result 3 stated in terms of polyphase representation) *Suppose g and h are two n-D lowpass filters with dilation Λ, and h is interpolatory. Let $\mathtt{G}(\omega)$ and $\mathtt{H}(\omega)$ be the polyphase representation of g and h with length $q = |\det \Lambda|$, and let $\tilde{\mathtt{G}}(\omega)$ and $\tilde{\mathtt{H}}(\omega)$ be the subvectors of $\mathtt{G}(\omega)$ and $\mathtt{H}(\omega)$ of length $q-1$, respectively, obtained by removing the first entry. Then the following two polyphase matrices*

$$\mathtt{A}(\omega) := \begin{bmatrix} \mathtt{G}_{\nu_0}(\omega) + q\,\mathtt{B}(\omega) & \tilde{\mathtt{G}}(\omega) \\ -q\,\tilde{\mathtt{H}}(\omega) & \mathtt{I}_{q-1} \end{bmatrix} \qquad \mathtt{S}(\omega) := \begin{bmatrix} \frac{1}{q} & -\frac{1}{q}\,\tilde{\mathtt{G}}(\omega) \\ \tilde{\mathtt{H}}(\omega) & \frac{1}{q}\mathtt{I}_{q-1} - \tilde{\mathtt{H}}(\omega)\tilde{\mathtt{G}}(\omega) \end{bmatrix} \tag{3.17}$$

satisfy $\mathtt{S}(\omega)\mathtt{A}(\omega) = (1/q)\mathtt{I}_q$, where $\mathtt{B}(\omega) := 1/q - \mathtt{G}(\omega)\mathtt{H}(\omega)$.

3.5.4 Proof of Result 3 in section §3.3.1

We want to show that τ, τ^{d}, t_ν and t_ν^{d}, $\nu \in \Gamma'$, in Result 3, satisfy the following identity (cf. (3.3) in Section §3.1.2)

$$\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega) + \sum_{\nu\in\Gamma'} \overline{t_\nu(\omega+\gamma)} t_\nu^{\mathrm{d}}(\omega) = \delta_{\gamma,0} = \begin{cases} 1, & \text{if } \gamma = 0, \\ 0, & \text{if } \gamma \in \Gamma^*\backslash 0. \end{cases}$$

By substituting the masks τ, τ^{d}, t_ν and t_ν^{d}, $\nu \in \Gamma'$, in Result 3, we get

$$\begin{aligned}
&\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega) + \sum_{\nu\in\Gamma'} \overline{t_\nu(\omega+\gamma)} t_\nu^{\mathrm{d}}(\omega) \\
=&\left(\overline{\widehat{g}(\omega+\gamma)} + \left(1 - \sum_{\tilde\gamma\in\Gamma^*} \overline{\widehat{g}(\omega+\tilde\gamma+\gamma)}\, \widehat{h}(\omega+\tilde\gamma+\gamma)\right)\right)\widehat{h}(\omega) \\
&+ \sum_{\nu\in\Gamma'} \left(e^{i(\omega+\gamma)\cdot\nu} - q(h(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)\right)\left(\frac{1}{q}e^{-i\omega\cdot\nu} - \overline{(g(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)}\, \widehat{h}(\omega)\right) \\
=&\overline{\widehat{g}(\omega+\gamma)}\, \widehat{h}(\omega) + \widehat{h}(\omega) - \sum_{\tilde\gamma\in\Gamma^*} \overline{\widehat{g}(\omega+\tilde\gamma+\gamma)}\, \widehat{h}(\omega+\tilde\gamma+\gamma)\widehat{h}(\omega) \\
&+\frac{1}{q}\sum_{\nu\in\Gamma} e^{i\gamma\cdot\nu} - \frac{1}{q} - \left(\sum_{\nu\in\Gamma} e^{i(\omega+\gamma)\cdot\nu}\, \overline{(g(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)}\, \widehat{h}(\omega) - \overline{g(\Lambda\cdot)\widehat{\ }(\Lambda^*\omega)}\, \widehat{h}(\omega)\right) \\
&- \left(\sum_{\nu\in\Gamma} e^{-i\omega\cdot\nu}(h(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega) - h(\Lambda\cdot)\widehat{\ }(\Lambda^*\omega)\right) \\
&+q\sum_{\nu\in\Gamma'} (h(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)\, \overline{(g(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)}\, \widehat{h}(\omega).
\end{aligned}$$

It is easy to see that the following identity is true:

$$\sum_{\nu\in\Gamma} e^{i\gamma\cdot\nu} = q\delta_{\gamma,0} = \begin{cases} q, & \text{if } \gamma = 0, \\ 0, & \text{if } \gamma \in \Gamma^*\backslash 0, \end{cases} \tag{3.18}$$

where $q = |\det\Lambda|$. Since h is interpolatory, we have

$$h(\Lambda\cdot)\widehat{\ }(\Lambda^*\omega) = \frac{1}{q}\,. \tag{3.19}$$

Then by using (3.18), (3.19), (3.15), and the fact that $(g(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega) = (g(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*(\omega+\gamma))$, for any $\nu \in \Gamma$, $\omega \in \mathbb{T}^n$, and $\gamma \in \Gamma^*$, we get

$$\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega) + \sum_{\nu\in\Gamma'} \overline{t_\nu(\omega+\gamma)} t_\nu^{\mathrm{d}}(\omega)$$

$$= \delta_{\gamma,0} - \sum_{\tilde{\gamma}\in\Gamma^*} \overline{\widehat{g}(\omega+\tilde{\gamma})}\, \widehat{h}(\omega+\tilde{\gamma})\widehat{h}(\omega) + q\sum_{\nu\in\Gamma}(h(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)\, \overline{(g(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)}\, \widehat{h}(\omega)$$

$$= \delta_{\gamma,0} - \left(\sum_{\tilde{\gamma}\in\Gamma^*} \overline{\widehat{g}(\omega+\tilde{\gamma})}\, \widehat{h}(\omega+\tilde{\gamma}) - q\sum_{\nu\in\Gamma}(h(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)\, \overline{(g(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)} \right) \widehat{h}(\omega).$$

Moreover, by (3.15), and the dual identity of (3.18):

$$\sum_{\gamma\in\Gamma^*} e^{i\gamma\cdot\nu} = q\delta_{\nu,0} = \begin{cases} q, & \text{if } \nu = 0, \\ \\ 0, & \text{if } \nu \in \Gamma'\backslash 0, \end{cases} \tag{3.20}$$

we have

$$\begin{aligned}
&\sum_{\gamma\in\Gamma^*} \overline{\widehat{g}(\omega+\gamma)}\, \widehat{h}(\omega+\gamma) \\
&= \sum_{\gamma\in\Gamma^*} \left(\sum_{\nu\in\Gamma} e^{i(\omega+\gamma)\cdot\nu}\, \overline{(g(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)} \right) \left(\sum_{\tilde{\nu}\in\Gamma} e^{-i(\omega+\gamma)\cdot\tilde{\nu}}\, (h(\tilde{\nu}+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega) \right) \\
&= \sum_{\tilde{\nu}\in\Gamma} \left(\sum_{\nu\in\Gamma} \Big(\sum_{\gamma\in\Gamma^*} e^{i\gamma\cdot(\nu-\tilde{\nu})} \Big) e^{i\omega\cdot\nu}\, \overline{(g(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)} \right) e^{-i\omega\cdot\tilde{\nu}} (h(\tilde{\nu}+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega) \\
&= q\sum_{\nu\in\Gamma}(h(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)\, \overline{(g(\nu+\Lambda\cdot))\widehat{\ }(\Lambda^*\omega)}.
\end{aligned}$$

Therefore,

$$\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega) + \sum_{\nu\in\Gamma'} \overline{t_\nu(\omega+\gamma)} t_\nu^{\mathrm{d}}(\omega) = \delta_{\gamma,0}.$$

This concludes the proof.

3.5.5 Proof of the identities in the decomposition algorithm in section §3.3.2

The *polyphase decomposition of a signal* y_j with respect to the dilation matrix $\Lambda = p\mathtt{I}_n$, with $q = |\det\Lambda| = p^n$, is defined as the Fourier series of $y_j(\nu+p\cdot)$, $\nu\in\Gamma$:

$$\mathtt{Y}_{\nu,j}(\omega) := (y_j(\nu+p\cdot))\widehat{\ }(\omega) = \frac{1}{q}\sum_{k\in\mathbb{Z}^n} y_j(\nu+pk)e^{-ik\cdot\omega}, \quad \omega\in\mathbb{T}^n,$$

and the *polyphase representation of a signal* y_j is defined as the column q-vector of the form

$$\mathtt{Y_j}(\omega) := [\mathtt{Y}_{\nu_0,j}(\omega), \mathtt{Y}_{\nu_1,j}(\omega), \cdots, \mathtt{Y}_{\nu_{q-1},j}(\omega)]^T, \quad \omega \in \mathbb{T}^n,$$

where $\nu_0 = 0$ and ν_j, $j = 1, \ldots, q-1$, are the ordered elements of the set Γ'. Let Y_{j-1} and $W_{\nu,j-1}$ be the Fourier series of coarse coefficients y_{j-1} and wavelet coefficients $w_{\nu,j-1}$, $\nu \in \Gamma'$, respectively,

$$\begin{aligned} Y_{j-1}(\omega) &:= \frac{1}{q} \sum_{k\in\mathbb{Z}^n} y_{j-1}(k) e^{-ik\cdot\omega}, \\ W_{\nu,j-1}(\omega) &:= \frac{1}{q} \sum_{k\in\mathbb{Z}^n} w_{\nu,j-1}(k) e^{-ik\cdot\omega}, \quad \nu \in \Gamma', \end{aligned}$$

for every $\omega \in \mathbb{T}^n$. Then an 1-level-down decomposition, in frequency domain, can be written as

$$\begin{bmatrix} Y_{j-1}(\omega) \\ \\ W_{j-1}(\omega) \end{bmatrix} = \mathtt{A}(\omega) \begin{bmatrix} \mathtt{Y}_{\nu_0,j}(\omega) \\ \\ \tilde{\mathtt{Y}}_j(\omega) \end{bmatrix},$$

where $W_{j-1}(\omega) := [W_{\nu_1,j-1}(\omega), \cdots, W_{\nu_{q-1},j-1}(\omega)]^T$ and $\tilde{\mathtt{Y}}_j(\omega)$ is a subvector of $\mathtt{Y}(\omega)$ of length $q-1$ obtained by removing the first entry.

A key observation, which is also part of the reason why the fast prime coset sum wavelet algorithms is fast, is that $\mathtt{A}(\omega)$ as defined in (3.17) can be decomposed into two triangular matrices:

$$\mathtt{A}(\omega) = \begin{bmatrix} 1 & \tilde{\mathtt{G}}(\omega) \\ \\ 0 & \mathtt{I}_{q-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \\ -q\,\tilde{\mathtt{H}}(\omega) & \mathtt{I}_{q-1} \end{bmatrix}.$$

Thus we can calculate $W_{j-1}(\omega)$ first, then use $W_{j-1}(\omega)$ to compute $Y_{j-1}(\omega)$ as follows,

$$W_{j-1}(\omega) = -q\,\tilde{\mathtt{H}}(\omega)\mathtt{Y}_{\nu_0,j}(\omega) + \tilde{\mathtt{Y}}_j(\omega), \tag{3.21}$$

$$Y_{j-1}(\omega) = \mathtt{Y}_{\nu_0,j}(\omega) + \tilde{\mathtt{G}}(\omega) W_{j-1}(\omega). \tag{3.22}$$

From these (3.21) and (3.22), we now derive Step `(i)` and `(ii)` in our decomposition algorithm.

From (3.21), (3.14) and Lemma 5, we know that, for any $\nu \in \Gamma'$,

$$\begin{aligned} W_{\nu,j-1}(\omega) &= -q\ \mathtt{H}_\nu(\omega)\mathtt{Y}_{\nu_0,j}(\omega) + \mathtt{Y}_{\nu,j}(\omega) \\ &= -q\ (h(\nu+p\cdot))\widehat{\ }(\omega)\mathtt{Y}_{\nu_0,j}(\omega) + \mathtt{Y}_{\nu,j}(\omega) \\ &= -\frac{p}{p-1}\sum_{l\in F_p'} e^{i\omega\cdot\frac{(\nu-\eta(l,\nu)l)}{p}}\Big(H(l+p\cdot)\Big)\widehat{\ }(\omega\cdot\eta(l,\nu))\mathtt{Y}_{\nu_0,j}(\omega) + \mathtt{Y}_{\nu,j}(\omega). \end{aligned}$$

Hence,

$$\begin{aligned} \frac{1}{p^n}\sum_{k\in\mathbb{Z}^n} w_{\nu,j-1}(k)e^{-ik\cdot\omega} = W_{\nu,j-1}(\omega) &= \frac{1}{p^n}\sum_{k\in\mathbb{Z}^n} y_j(pk+\nu)e^{-ik\cdot\omega} \\ -\frac{p}{p-1}\sum_{l\in F_p'} e^{i\omega\cdot\frac{(\nu-\eta(l,\nu)l)}{p}}\frac{1}{p}\sum_{m\in\mathbb{Z}} H(l+pm)e^{-im(\omega\cdot\eta(l,\nu))}&\frac{1}{p^n}\sum_{k'\in\mathbb{Z}^n} y_j(pk')e^{-ik'\cdot\omega}. \end{aligned}$$

Therefore,

$$\begin{aligned} &\sum_{k\in\mathbb{Z}^n} w_{\nu,j-1}(k)e^{-ik\cdot\omega} \\ &= \sum_{k\in\mathbb{Z}^n} y_j(pk+\nu)e^{-ik\cdot\omega} \\ &\quad -\frac{1}{p-1}\sum_{l\in F_p'}\sum_{m\in\mathbb{Z}}\sum_{k'\in\mathbb{Z}^n} e^{i\omega\cdot\frac{(\nu-\eta(l,\nu)l)}{p}}e^{-im(\omega\cdot\eta(l,\nu))}e^{-ik'\cdot\omega}H(l+pm)y_j(pk') \\ &= \sum_{k\in\mathbb{Z}^n} y_j(pk+\nu)e^{-ik\cdot\omega} \\ &\quad -\frac{1}{p-1}\sum_{l\in F_p'}\sum_{m\in\mathbb{Z}}\sum_{k\in\mathbb{Z}^n} e^{-ik\cdot\omega}H(l+pm)y_j(pk+\nu-\eta(l,\nu)(pm+l)) \\ &= \sum_{k\in\mathbb{Z}^n}\left(y_j(pk+\nu) - \frac{1}{p-1}\sum_{l\in F_p'}\sum_{m\in\mathbb{Z}} H(l+pm)y_j(pk+\nu-\eta(l,\nu)(pm+l))\right)e^{-ik\cdot\omega} \\ &= \sum_{k\in\mathbb{Z}^n}\left(y_j(pk+\nu) - \frac{1}{p-1}\sum_{l\in F_p'}\sum_{m\equiv l} H(m)y_j(pk+\nu-\eta(l,\nu)(m))\right)e^{-ik\cdot\omega}, \end{aligned}$$

which in turn implies that we have for any $k \in \mathbb{Z}^n$ and $\nu \in \Gamma'$,

$$w_{\nu,j-1}(k) \;=\; y_j(pk+\nu) - \frac{1}{p-1}\sum_{l\in F_p'}\sum_{m\equiv l} H(m)y_j(pk+\nu-\eta(l,\nu)m).$$

This is exactly Step (i) in our decomposition algorithm.

From (3.22), (3.16) and by Lemma 5 we know that

$$\begin{aligned}
Y_{j-1}(\omega) &= \mathtt{Y}_{\nu_0,j}(\omega) + \sum_{\nu\in\Gamma'} \mathtt{G}_\nu(\omega) W_{\nu,j-1}(\omega) \\
&= \mathtt{Y}_{\nu_0,j}(\omega) + \sum_{\nu\in\Gamma'} \overline{g(\nu+p\cdot)\widehat{\ }(\omega)}\; W_{\nu,j-1}(\omega) \\
&= \mathtt{Y}_{\nu_0,j}(\omega) + \sum_{\nu\in\Gamma'} \frac{1}{(p-1)p^{n-1}} \sum_{l\in F_p'} e^{i\omega\cdot\frac{(\eta(l,\nu)l-\nu)}{p}}\; \overline{\Big(G(l+p\cdot)\Big)\widehat{\ }(\omega\cdot\eta(l,\nu))}\; W_{\nu,j-1}(\omega).
\end{aligned}$$

Hence,

$$\begin{aligned}
&\frac{1}{p^n}\sum_{k\in\mathbb{Z}^n} y_{j-1}(k)e^{-ik\cdot\omega} = Y_{j-1}(\omega) = \frac{1}{p^n}\sum_{k\in\mathbb{Z}^n} y_j(pk)e^{-ik\cdot\omega} + \\
&\sum_{\nu\in\Gamma'}\frac{1}{(p-1)p^n}\sum_{l\in F_p'} e^{i\omega\cdot\frac{(\eta(l,\nu)l-\nu)}{p}} \sum_{m\in\mathbb{Z}^n} G(l-pm)e^{-im(\omega\cdot\eta(l,\nu))}\frac{1}{p^n}\sum_{k'\in\mathbb{Z}^n} w_{\nu,j-1}(k')e^{-ik'\cdot\omega}.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\sum_{k\in\mathbb{Z}^n} y_{j-1}(k)e^{-ik\cdot\omega} \\
&= \sum_{k\in\mathbb{Z}^n} y_j(pk)e^{-ik\cdot\omega} \\
&\quad + \frac{1}{(p-1)p^n}\sum_{\nu\in\Gamma'}\sum_{l\in F_p'}\sum_{m\in\mathbb{Z}^n}\sum_{k'\in\mathbb{Z}^n} e^{i\omega\cdot\frac{(\eta(l,\nu)l-\nu)}{p}} e^{-im(\omega\cdot\eta(l,\nu))}e^{-ik'\cdot\omega}G(l-pm)w_{\nu,j-1}(k') \\
&= \sum_{k\in\mathbb{Z}^n} y_j(pk)e^{-ik\cdot\omega} \\
&\quad + \frac{1}{(p-1)p^n}\sum_{\nu\in\Gamma'}\sum_{l\in F_p'}\sum_{m\in\mathbb{Z}^n}\sum_{k\in\mathbb{Z}^n} e^{-ik\cdot\omega}G(l-pm)w_{\nu,j-1}(k-\frac{\nu-\eta(l,\nu)l}{p}-\eta(l,\nu)m) \\
&= \sum_{k\in\mathbb{Z}^n}\left(y_j(pk) + \frac{1}{(p-1)p^n}\sum_{\nu\in\Gamma'}\sum_{l\in F_p'}\sum_{m\equiv l} G(m)w_{\nu,j-1}(k-\frac{\nu-\eta(l,\nu)m}{p})\right)e^{-ik\cdot\omega},
\end{aligned}$$

As a result, we have, for any $k \in \mathbb{Z}^n$,

$$y_{j-1}(k) = y_j(pk) + \frac{1}{(p-1)p^n} \sum_{\nu \in \Gamma'} \sum_{l \in F'_p} \sum_{m \equiv l} G(m) w_{\nu,j-1}(k - \frac{\nu - \eta(l,\nu)m}{p}).$$

This is exactly Step `(ii)` in our decomposition algorithm.

Chapter 4

Multi-D Wavelet Filter Bank Design using Quillen-Suslin Theorem for Laurent Polynomials

4.1 Introduction

The main objective of this paper is to present a new approach for constructing nonseparable multidimensional (multi-D) non-redundant wavelet filter banks (FBs). Constructing wavelet FBs is often reduced to solving a matrix equation with Laurent polynomial entries [46]. Connecting wavelet FBs with the Laurent polynomial matrices is usually done by the polyphase representation [77]. The key idea for our method is to decompose the z-transform of filters using, instead of the usual polyphase representation, a special type of valid (generalized) polyphase representation [79], which we obtain from the Quillen-Suslin

Theorem for Laurent polynomials. This new representation allows us to use the matrix analysis techniques that were not available for the usual polyphase representation.

Quillen-Suslin Theorem (or unimodular completion), a celebrated theorem in Algebraic Geometry, states that a unimodular matrix with polynomial entries can be completed to a square polynomial matrix of determinant 1. This result was extended by R. G. Swan to unimodular matrices with Laurent polynomial entries [80].

While there have been several uses of unimodular completion in constructing multi-D FBs [47], constructing multi-D wavelet FBs using unimodular completion is mostly done by imposing additional constraints after multi-D FBs are constructed. Our method is different from these existing methods in that it gives an algorithm to construct multi-D wavelet FBs more readily. Our method provides an algorithm for constructing a wavelet FB from a single lowpass filter so that its vanishing moments are at least as many as the accuracy number of the lowpass filter.

The wavelet representation, along with Fourier representation, has been one of the most effective data representations. Constructing 1-D wavelets is well understood by now, but the situation is quite different for multi-D case. The most commonly used method for constructing multi-D wavelets is the tensor product, but the resulting wavelets have many unavoidable limitations. Many researches on constructing non-tensor-based multi-D wavelet FBs or wavelets have been performed [4, 7–20, 29, 35, 36, 44, 49]. Drawbacks of existing non-tensor-based multi-D wavelet constructions include the following. Many of the existing methods work only for low spatial dimensions and cannot be easily extended to higher dimensions. Others assume that the lowpass filters or refinable functions satisfy

additional conditions such as interpolatory condition. Our construction method presents some advantages over these existing methods of multi-D wavelet construction. It works for any spatial dimension and for any sampling matrix. Furthermore, it does not require the initial lowpass filters to satisfy any additional assumption such as interpolatory condition.

We now outline the rest of our paper. In Section §4.2, we briefly review some technical background about wavelet FBs, unimodular completion and other relevant concepts. In Section §4.3, we present our main results together with examples illustrating our findings. We summarize our results and provide outlooks in Section §4.4. Appendix §4.5 contains some technical proofs.

4.2 Preliminaries

4.2.1 Wavelet filter banks and their polyphase representation

Let Λ be an $n \times n$ integer sampling or dilation matrix. By definition, this means that Λ is an integer matrix and its spectrum lies outside the closed unit disc. Throughout the paper, we use q to denote the magnitude of $\det \Lambda$, i.e. $q := |\det \Lambda|$.

A Laurent trigonometric polynomial is typically referred to as a *mask*, and a mask τ with $\tau(0) = \sqrt{q}$ and $\tau(0) = 0$ as a *refinement mask* and *wavelet mask*, respectively. It is well known that refinement masks can be used to obtain refinable functions used in wavelet construction via the cascade algorithm (or subdivision scheme) [41] and, together with wavelet masks, they can be used to construct wavelet systems in $L^2(\mathbb{R}^n)$ [1]. We recall that a filter $f : \mathbb{Z}^n \to \mathbb{R}$ is *associated with* a mask τ if τ is the Fourier transform of f. A

filter f is called *lowpass* or *refinement* if

$$\sum_{k\in\mathbb{Z}^n} f(k) = \sqrt{q},$$

and *highpass* or *wavelet* if

$$\sum_{k\in\mathbb{Z}^n} f(k) = 0.$$

In this paper we consider only the finite impulse response (FIR) filters. A FB consists of the analysis bank and the synthesis bank, which are collections of finite number, say p, of FIR filters linked by downsampling and upsampling operators, respectively, with the sampling matrix Λ [3]. We refer to a filter from the analysis bank as an analysis filter and a filter from the synthesis bank as a synthesis filter. We consider only the FBs that satisfy the perfect reconstruction condition, which implies $p \geq q$. We are interested in the FB for which each of its analysis and synthesis banks has exactly one lowpass filter and the rest of them are all highpass filters. We refer to such a FB as a *wavelet* FB. A FB is called *critically sampled* or *non-redundant* if $p = q$ and *oversampled* or *redundant* otherwise. Designing non-redundant wavelet FBs is an important problem since it leads to the construction of wavelet bases under well-understood constraints [1–3].

We recall that for a filter f, the number of zeros of the Fourier transform of f at $\omega = 0$ is referred to as the *number of (discrete) vanishing moments of the filter* f [36]. Thus, a filter f is highpass if and only if f has at least one vanishing moment. We say that a wavelet FB has $s \in \mathbb{N}$ vanishing moments if the minimum of all its highpass filters' vanishing moments is s.

We use Γ to denote a complete set of representatives of the distinct cosets of the quotient group $\mathbb{Z}^n/\Lambda\mathbb{Z}^n$ containing 0, and Γ^* to denote a complete set of representatives of

the distinct cosets of $2\pi(((\Lambda^*)^{-1}\mathbb{Z}^n)/\mathbb{Z}^n)$ containing 0. Throughout this paper, for a matrix M, M^* is used to denote its conjugate transpose. We note that both the sets Γ and Γ^* have $q = |\det \Lambda|$ elements. For example, for the 2-D dyadic dilation matrix $\Lambda = 2\mathtt{I}_2$, the sets $\Gamma = \{(0,0),(1,0),(0,1),(1,1)\}$ and $\Gamma^* = \{(0,0),(\pi,0),(0,\pi),(\pi,\pi)\}$ can be used. We also use the notation

$$\nu_0 = 0, \nu_1, \cdots, \nu_{q-1}$$

to denote the elements of Γ.

The concept of polyphase decomposition is to transform a filter or a signal into q filters or signals running at the sampling rate $1/q$ [77]. For a given FB, let h be an analysis filter, and g a synthesis filter. Then the polyphase decomposition of h (respectively, g) is a set of q filters h_ν, $\nu \in \Gamma$, (respectively, g_ν, $\nu \in \Gamma$) that are defined as

$$h_\nu(m) := h(\Lambda m - \nu), \quad g_\nu(m) := g(\Lambda m + \nu), \quad \forall m \in \mathbb{Z}^n.$$

The z-transform ([81]) $Y(z)$ of a filter $y : \mathbb{Z}^n \to \mathbb{R}$ is defined as

$$Y(z) := \mathcal{Z}\{y\} := \sum_{m\in\mathbb{Z}^n} y(m) z^{-m}$$

where for $z = [z_1, \ldots, z_n]^T \in \mathbb{C}^n \backslash \{0\}$ with $|z| = 1$ and $m = [m_1, \ldots, m_n]^T \in \mathbb{Z}^n$, z^m is defined to be $\prod_{j=1}^n z_j^{m_j}$. Here and below, T is used to represent the matrix transpose. We note that $Y(e^{i\omega})$, $\omega \in \mathbb{T}^n$, is the Fourier transform of y. We let $\mathtt{1} := [1, \cdots, 1]^T$ be the vector of ones. The z-transforms of the filters h and g can be written as

$$H(z) = \sum_{\nu\in\Gamma} z^\nu H_\nu(z^\Lambda), \quad G(z) = \sum_{\nu\in\Gamma} z^{-\nu} G_\nu(z^\Lambda) \tag{4.1}$$

where G_ν and H_ν are the z-transforms of g_ν and h_ν, and $z^\Lambda := [z^{\Lambda_1}, \ldots, z^{\Lambda_n}]^T$ with the column vectors $\Lambda_1, \ldots, \Lambda_n$ of Λ. The *polyphase representation* of the filters h and g are

defined as

$$\mathtt{H}(z) := [H_{\nu_0}(z), H_{\nu_1}(z), \ldots, H_{\nu_{q-1}}(z)], \quad \mathtt{G}(z) := [G_{\nu_0}(z), G_{\nu_1}(z), \ldots, G_{\nu_{q-1}}(z)]^T.$$

The polyphase representation of analysis and synthesis parts of a FB can be represented by a $p \times q$ matrix $\mathtt{A}(z)$ and a $q \times p$ matrix $\mathtt{S}(z)$, respectively, where p is the number of filters in each bank. In this case, the row vectors of $\mathtt{A}(z)$ represent the polyphase representation of analysis filters, and the column vectors of $\mathtt{S}(z)$ represent the polyphase representation of synthesis filters. Then the perfect reconstruction condition of the FB becomes $\mathtt{S}(z)\mathtt{A}(z) = \mathtt{I}_q$, with $p \geq q$. For non-redundant FBs, the polyphase matrices $\mathtt{A}(z)$ and $\mathtt{S}(z)$ should be $q \times q$ square matrices.

Finally we briefly review the valid polyphase representation [79] in our context. If we define $v(z) := [1, z^{\nu_1}, \cdots, z^{\nu_{q-1}}]^T$ to be the usual polyphase basis, then from (4.1), we see that the z-transform of h can be written as

$$H(z) = \mathtt{H}(z^\Lambda)v(z).$$

We recall that $u(z) := \mathtt{M}(z^\Lambda)v(z)$ is called a *valid polyphase basis* if and only if $\mathtt{M}(z)$ is an invertible matrix, i.e. $\mathtt{M}(z) \in \mathrm{GL}_q(\mathbb{R}[z^{\pm 1}])$. Then the z-transform of the filter can be written using the new basis,

$$H(z) = \mathtt{H}^u(z^\Lambda)u(z),$$

where

$$\mathtt{H}^u(z) := \mathtt{H}(z)[\mathtt{M}(z)]^{-1}$$

is called the *valid (generalized) polyphase representation* of the filter h with respect to the basis $u(z)$.

4.2.2 Unimodular vector completion and its use in FB design

Let k be a field and let $k[z^{\pm 1}]$ be the Laurent polynomial ring, consisting of all Laurent polynomials in $z = [z_1, \ldots, z_n]^T$ with coefficients in k. A vector $\mathbf{v} = [v_1, \ldots, v_n]$ with Laurent polynomial entries is called *unimodular* if its entries generate 1, i.e. there exist Laurent polynomials $g_1, \ldots, g_n$ such that $v_1 g_1 + \cdots + v_n g_n = 1$. In general, a matrix with Laurent polynomial entries is called a *unimodular matrix* if its maximal minors generate 1.

In 1955, Jean Pierre Serre made a conjecture regarding vector bundles over an affine space [82]. This problem became a daunting task for many mathematicians, and was fully solved only in 1976, 20 years after the question was raised. Serre's conjecture, which is now known as the Quillen-Suslin Theorem ([83, 84]) after the two mathematicians who independently solved this long standing problem, asserts that any unimodular matrix over a polynomial ring can be completed to an invertible square matrix, i.e. a square matrix of nonzero constant determinant. And in 1978, R.G. Swan [80] extended this result to the case of Laurent polynomial rings.

Theorem 4 (Unimodular Completion, or Quillen-Suslin Theorem for Laurent polynomials) *Let* $\mathbf{A}$ *be a* $p \times q$ *unimodular matrix,* $p \geq q$*, with Laurent polynomial entries. Then* $\mathbf{A}$ *can be completed to a square* $p \times p$ *unimodular matrix* $\bar{\mathbf{A}} \in \mathrm{GL}_p(k[z^{\pm 1}])$ *by adding* $p - q$ *columns to the matrix* $\mathbf{A}$*.* ■

The polyphase representation of a FB consists of the Laurent polynomials in z with real coefficients, which allows many concepts and results in FB design to be stated in terms of these Laurent polynomials. For example, we recall that the two polyphase lowpass filters $\mathtt{H}(z)$ (analysis) and $\mathtt{G}(z)$ (synthesis), or the associated filters h and g, are called *biorthogonal*

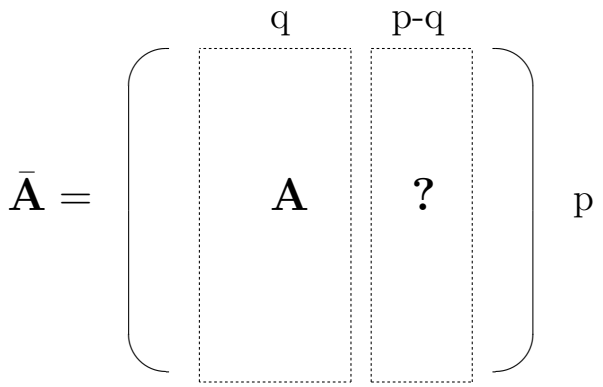


Figure 4.1: Unimodular completion of $\mathbf{A}$ to $\bar{\mathbf{A}}$

if $\mathtt{H}(z)\mathtt{G}(z) = 1$, which is equivalent to the row vector $\mathtt{H}(z)$ or the column vector $\mathtt{G}(z)$ being *unimodular*. In such a case, $\mathtt{G}(z)$ (respectively, g) is called a *dual* of $\mathtt{H}(z)$ (respectively, h). Hilbert's Nullstellensatz ([85]) for the Laurent polynomial ring $\mathbb{R}[z^{\pm 1}]$ says that a given row vector $\mathtt{H}(z) = [H_{\nu_0}(z), H_{\nu_1}(z), \ldots, H_{\nu_{q-1}}(z)]$ is unimodular if and only if the Laurent polynomials $H_\nu(z)$, $\nu \in \Gamma$, do not have a nonzero complex common root. Therefore, for a given polyphase analysis lowpass filter $\mathtt{H}(z)$, a dual polyphase synthesis filter $\mathtt{G}(z)$ exists if and only if the components of $\mathtt{H}(z)$ do not have a nonzero complex common root. For a given a unimodular polyphase analysis lowpass filter $\mathtt{H}(z)$, Gröbner bases techniques ([86]) can be used to find a particular dual polyphase synthesis lowpass filter, as well as the most general form of dual lowpass filters.

Our method is based on the following special case of the unimodular completion over Laurent polynomial rings:

Result 5 (Unimodular vector completion) *Let* $\mathtt{F}(z) \in \mathbb{R}[z^{\pm 1}]^q$ *be a unimodular column vector of length* q. *Then there exists an invertible* $q \times q$ *matrix* $\mathtt{K}(z) \in \mathrm{GL}_q(\mathbb{R}[z^{\pm 1}])$ *such that* $\mathtt{K}(z)\mathtt{F}(z) = [1, 0, \ldots, 0]^T$. ■

While the original proofs of Quillen-Suslin Theorem were nonconstructive, algorithmic proofs were studied in [87–89]. By using these algorithms, given a unimodular polynomial vector $\mathtt{F}(z)$, one can compute a companion unimodular polynomial matrix $\mathtt{K}(z)$ in Result 5. This algorithm was extended to unimodular Laurent polynomial matrices in [90], which was implemented as a part of the Maple package *QuillenSuslin* by Anna Fabiańska (see http://wwwb.math.rwth-aachen.de/QuillenSuslin/).

There have been many studies on the design of multi-D FBs using unimodular completion (cf. Section §4.1), but there was little success in developing a simple construction method for wavelet FBs, not just FBs. In other words, how one can make sure the resulting FB to have a certain number of vanishing moments, without much work, has been a remaining challenge for the most part. Our approach in this paper provides an answer toward this direction.

It is well known that (see, for example, [36]) the number of vanishing moments of the non-redundant wavelet FB is at least s if the accuracy numbers of its lowpass filters are at least s. We recall that for a given lowpass filter f, the number of zeros of the Fourier transform of f at $\omega \in \Gamma^* \backslash \{0\}$ is referred to as the *accuracy number* [3]. This number determines the maximum degree of polynomials that can be reproduced by the filter f and it is closely related with the Strang-Fix order in the wavelet theory [91]. When a wavelet FB gives rise to a wavelet system in $L^2(\mathbb{R}^n)$, the number of vanishing moments of the wavelet system is completely determined by the (discrete) vanishing moments of the wavelet FB. Therefore, for constructing multi-D wavelet bases with a certain number of vanishing moments, we can start from the two biorthogonal lowpass filters with prescribed

accuracy numbers. Unfortunately, this too is not easy in general and requires great care in the construction process. Our result (Corollary 4) presented in the next section provides a solution to this problem.

4.3 Construction of multi-D wavelet FBs using Quillen-Suslin theorem

In this section, we present a new method for constructing multi-D wavelets using the Quillen-Suslin Theorem over Laurent polynomials. From this method, algorithms for constructing a non-redundant multi-D wavelet FB just from a single lowpass filter can be obtained. The motivation and the main idea of our method is presented in Section §4.3.1, the main results are shown in Section §4.3.2, and the algorithms are shown in Section §4.3.3.

4.3.1 Motivation

Many of the existing construction methods for multi-D wavelet systems ([4, 29, 35, 36, 44, 49]) assume that at least one of the lowpass filters is interpolatory. We recall that a lowpass filter f is *interpolatory* if

$$f(0) = \frac{1}{\sqrt{q}} \text{ and } f(\Lambda m) = 0, \forall m \in \mathbb{Z}^n \backslash \{0\}.$$

Equivalently, the polyphase lowpass filter $\mathtt{F}(z)$ is interpolatory if its first component satisfies

$$F_{\nu_0}(z) = \frac{1}{\sqrt{q}}.$$

It is easy to see that every polyphase interpolatory lowpass filter $\mathtt{F}(z)$ is unimodular, since the dual vector can be chosen so that its first component is $\sqrt{q}$ and the rest are all zero.

The Laplacian pyramid (LP) representation ([45]) has been used in many image processing applications [92–94]. In the LP algorithms, if the interpolatory lowpass filter h is used for analysis and the "lazy" interpolatory ([95]) lowpass filter g is used for synthesis as its dual, then we have ([78])

$$\mathtt{H}(z) = [\frac{1}{\sqrt{q}}, H_{\nu_1}(z), \cdots, H_{\nu_{q-1}}(z)], \quad \mathtt{G}(z) = [\sqrt{q}, 0, \cdots, 0]^T$$

and

$$\begin{bmatrix} \mathtt{G}(z) & \mathtt{I}_q \end{bmatrix} \begin{bmatrix} \mathtt{H}(z) \\ \\ \mathtt{I}_q - \mathtt{G}(z)\mathtt{H}(z) \end{bmatrix} = \mathtt{I}_q.$$

Although the above matrices can be considered as a polyphase representation of a redundant FB, it is clear that this FB is not a wavelet FB as the synthesis filters associated with the column vectors of the polyphase matrix $\mathtt{I}_q$ do not have any vanishing moment. A new method called the interpolatory effortless critical representation of LP is proposed in order to transform these LP-based, *redundant non-wavelet*, FBs to *non-redundant wavelet* FBs in a remarkably simple way [44]. This new method provides a way to construct non-redundant wavelet FBs for any dimension and any dilation. A critical assumption for this method is that $\mathtt{H}(z)$ has to be essentially interpolatory (see (23) in [44] for a precise statement of the assumption).

A closer look at the interpolatory lowpass filter reveals that not only its polyphase representation $\mathtt{H}(z)$ is unimodular, but also it has a dual with a unit in at least one of its components. We recall that an element in a ring is called a *unit* if its multiplicative inverse lies in the ring. Scrutinizing the techniques used in [44] shows that many arguments used there rely on this "nice" property of analysis interpolatory lowpass filters. Therefore it is not clear how to directly apply them to more general analysis lowpass filters.

On the other hand, we notice that many techniques used in [44] do not care whether the Laurent polynomial matrices come from the polyphase representation or not. The key idea of our new construction method is to decompose the z-transform of filters using a special type of the valid polyphase representations obtained by unimodular vector completion over Laurent polynomial rings. In some sense, this can be understood as a change of basis, from the usual polyphase basis to the valid polyphase basis, in the Laurent polynomial ring. In the next subsection, we show exactly how this new representation is obtained.

4.3.2 Main results

Our new construction method relies on Result 5. In fact, the following slightly modified version of Result 5 is sufficient for the arguments in the proof and it gives more flexibility in the construction process.

Corollary 2 (A slightly modified version of Result 5) *Let* $\mathtt{F}(z) \in \mathbb{R}[z^{\pm 1}]^q$ *be a unimodular column vector of length* q*. Then there exists an invertible* $q \times q$ *matrix* $\mathtt{T}(z) \in \mathrm{GL}_q(\mathbb{R}[z^{\pm 1}])$ *such that* $\mathtt{T}(z)\mathtt{F}(z)$ *is a unimodular column vector that has a unit in at least one of its components.* ■

Our main theorem is placed below. It provides the theory and the algorithm to construct a non-redundant wavelet FB from a lowpass filter whose polyphase representation is unimodular. It uses Corollary 2 and part of the arguments used to prove some results (Theorem 1 and 2) in [44]. It is also a variant of a result[1] (Theorem 1) in [96].

[1]While the statement of Theorem 1 in [96] is correct, the proof presented there turns out to contain an error.

Theorem 5 *Let h be a lowpass filter with positive accuracy. If its polyphase representation $\mathtt{H}(z)$ as a row vector is unimodular, then there exists a non-redundant wavelet FB whose analysis lowpass filter is h.* ■

Proof 7 *Since $\mathtt{H}(z) = [H_{\nu_0}(z), \ldots, H_{\nu_{q-1}}(z)]$ is unimodular, there exists*

$$\mathtt{F}(z) = [F_{\nu_0}(z), \ldots, F_{\nu_{q-1}}(z)]^T$$

such that

$$\mathtt{H}(z)\mathtt{F}(z) = H_{\nu_0}(z)F_{\nu_0}(z) + \ldots + H_{\nu_{q-1}}(z)F_{\nu_{q-1}}(z) = 1.$$

Thus $\mathtt{F}(z)$ is also unimodular. By Corollary 2, there exists an invertible $q \times q$ square matrix $\mathtt{T}(z)$ such that $\mathtt{T}(z)\mathtt{F}(z)$ is a unimodular vector with a unit in at least one of its components. Without loss of generality, we assume the first component of $\mathtt{T}(z)\mathtt{F}(z)$ is a unit.

Let g be another lowpass filter with positive accuracy that can possibly be different from h, and let $\mathtt{G}(z) := [G_{\nu_0}(z), G_{\nu_1}(z), \ldots, G_{\nu_{q-1}}(z)]^T$ be its synthesis polyphase representation.

From the discussion at the end of Section §4.2.1, we see that the z-transform of h, f and g can be written as

$$H(z) = \mathtt{H}(z^\Lambda)v(z), \quad F(z) = v(z)^*\mathtt{F}(z^\Lambda), \quad G(z) = v(z)^*\mathtt{G}(z^\Lambda),$$

where $v(z) = [1, z^{\nu_1}, \cdots, z^{\nu_{q-1}}]^T$ is the usual polyphase basis as before, and $v(z)^* := v(z^{-1})^T$ is the conjugate transpose of $v(z)$.

We take the approach in [79] but extend it slightly by allowing two different valid polyphase bases for analysis and synthesis filters. More precisely, using the above invertible matrix $\mathtt{T}(z)$, we define a new pair of valid polyphase bases $u(z) := \mathtt{T}(z^\Lambda)v(z)$ and $w(z) :=$

$[\mathtt{T}(z^\Lambda)^*]^{-1}v(z)$, *and use them instead of the usual basis* $v(z)$ *to represent the* z*-transform of the analysis and the synthesis filters, respectively. For example,*

$$H(z) = \mathtt{H}^u(z^\Lambda)u(z), \quad F(z) = w(z)^*\mathtt{F}^w(z^\Lambda), \quad G(z) = w(z)^*\mathtt{G}^w(z^\Lambda),$$

where

$$\mathtt{H}^u(z) := \mathtt{H}(z)[\mathtt{T}(z)]^{-1}, \quad \mathtt{F}^w(z) := \mathtt{T}(z)\mathtt{F}(z), \quad \mathtt{G}^w(z) := \mathtt{T}(z)\mathtt{G}(z)$$

are the valid polyphase representation of h, f *and* g *with respect to the new valid polyphase basis pair* $(u(z), w(z))$.

Then from the fact that $\mathtt{F}^w(z)$ *is a particular dual to* $\mathtt{H}^u(z)$, *i.e.* $\mathtt{H}^u(z)\mathtt{F}^w(z) = \mathtt{H}(z)\mathtt{F}(z) = 1$, *we see that any column vector of the form* $\mathtt{G}^w(z) + \mathtt{F}^w(z)(1 - \mathtt{H}^u(z)\mathtt{G}^w(z))$ *is also dual to* $\mathtt{H}^u(z)$. *In fact, it is easy to see that the matrix identity*

$$\begin{bmatrix} \mathtt{G}^w(z) + \mathtt{F}^w(z)(1 - \mathtt{H}^u(z)\mathtt{G}^w(z)) & \mathtt{I}_q - \mathtt{F}^w(z)\mathtt{H}^u(z) \end{bmatrix} \begin{bmatrix} \mathtt{H}^u(z) \\ \mathtt{I}_q - \mathtt{G}^w(z)\mathtt{H}^u(z) \end{bmatrix} = \mathtt{I}_q \tag{4.2}$$

always holds true.

Since $F^w_{\nu_0}(z)$ *is assumed to be a unit, if we let the* $(q+1)\times(q+1)$ *reduction matrix* $\mathtt{R}(z)$ *to be*

$$\mathtt{R}(z) := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & c(z)F^w_{\nu_0}(z) & & & \\ 0 & c(z)F^w_{\nu_1}(z) & 1 & & \\ 0 & \vdots & & \ddots & \\ 0 & c(z)F^w_{\nu_{q-1}}(z) & & & 1 \end{bmatrix}$$

with any unit $c(z)$ *in the Laurent polynomial ring* $\mathbb{R}[z^{\pm 1}]$, *then the second column of*

$$\begin{bmatrix} \mathtt{G}^w(z) + \mathtt{F}^w(z)(1 - \mathtt{H}^u(z)\mathtt{G}^w(z)) & \mathtt{I}_q - \mathtt{F}^w(z)\mathtt{H}^u(z) \end{bmatrix} \mathtt{R}(z)$$

becomes a zero column vector. Since the reduction matrix $\mathtt{R}(z)$ *is invertible, i.e.* $\mathtt{R}(z) \in \mathrm{GL}_{q+1}(\mathbb{R}[z^{\pm 1}])$*, by inserting* $\mathtt{R}(z)[\mathtt{R}(z)]^{-1}$ *between the two matrices in the left-hand side of (4.2), we get*

$$[\mathtt{G}^w(z)+\mathtt{F}^w(z)(1-\mathtt{H}^u(z)\mathtt{G}^w(z)), \mathtt{I}_q-\mathtt{F}^w(z)\mathtt{H}^u(z)]\mathtt{R}(z)[\mathtt{R}(z)]^{-1} \begin{bmatrix} \mathtt{H}^u(z) \\ \mathtt{I}_q - \mathtt{G}^w(z)\mathtt{H}^u(z) \end{bmatrix} = \mathtt{I}_q \quad (4.3)$$

By letting $\mathtt{S}(z)$ *be the* $q \times q$ *matrix obtained by deleting the second column of the product of the first two matrices in the left-hand side, and* $\mathtt{A}(z)$ *be the* $q \times q$ *matrix obtained by deleting the second row of the product of the last two matrices in the left-hand side, we get a non-redundant FB with* $\mathtt{S}(z)\mathtt{A}(z) = \mathtt{I}_q$*.*

Since the first row of $[\mathtt{R}(z)]^{-1}$ *is* $[1, 0, \cdots, 0]$*, the first row of the analysis polyphase matrix* $\mathtt{A}(z)$ *is* $\mathtt{H}^u(z)$*, which in turn implies that the analysis lowpass filter is* h *in the above non-redundant FB. In order to finish the proof, we need to show that the non-redundant FB obtained above is a wavelet FB. It suffices to show that both the analysis lowpass filter* h *and the synthesis lowpass filter, say* d*, have positive accuracy (cf. Section §4.2.2). Since* h *has positive accuracy by the assumption, we only need to show that* d *has positive accuracy. Since its polyphase representation satisfies*

$$\mathtt{D}(z) = [\mathtt{T}(z)]^{-1}(\mathtt{G}^w(z) + \mathtt{F}^w(z)(1 - \mathtt{H}^u(z)\mathtt{G}^w(z))) = \mathtt{G}(z) + \mathtt{F}(z)(1 - \mathtt{H}(z)\mathtt{G}(z)),$$

and since both h *and* g *are assumed to have positive accuracy, we have* $\mathtt{D}(1) = \mathtt{G}(1)+\mathtt{F}(1)(1-\mathtt{H}(1)\mathtt{G}(1)) = \frac{1}{\sqrt{q}}[1, \ldots, 1]^T + \mathtt{F}(1)(1 - \frac{1}{\sqrt{q}}[1, \ldots, 1]\frac{1}{\sqrt{q}}[1, \ldots, 1]^T) = \frac{1}{\sqrt{q}}[1, \ldots, 1]^T$*, from which we can conclude that* d *also has positive accuracy (cf. Result 2 in [44]).* ■

Remark 1: Although we stated Theorem 5 for the case when the lowpass filter is used for the analysis, a similar statement can be made for the synthesis lowpass filter. ■

Remark 2: It is easy to see that the converse of the statement of Theorem 5 is also true.

■

Although the construction method developed in the above theorem works for any dimension and for any dilation, it is especially useful for the wavelet construction in multi-D setting as this is where the problem gets more challenging. We now present 2-D examples to illustrate our findings. For simplicity, in all of our examples, we consider the dyadic dilation and choose $\Gamma = \{(0,0),(1,0),(0,1),(1,1)\}$.

Example 1 (2-D wavelet FB generated from an interpolatory lowpass filter). Let h be the lowpass filter associated with the bivariate piecewise-linear box spline $B_{1,1,1}$ based on the three directions $(1,0),(0,1)$, and $(1,1)$ (see [40] for the definition of box splines and their properties), i.e.

$$h: \quad \begin{array}{ccc} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \boxed{\frac{1}{2}} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \end{array}$$

Here and below, the number in the box represents the coefficient of the filter at the origin. Since h is interpolatory and its polyphase representation is

$$\mathtt{H}(z) = \begin{bmatrix} \frac{1}{2} & \frac{1}{4}z_1^{-1} + \frac{1}{4} & \frac{1}{4}z_2^{-1} + \frac{1}{4} & \frac{1}{4}z_1^{-1}z_2^{-1} + \frac{1}{4} \end{bmatrix},$$

we can choose $\mathtt{F}(z) = \begin{bmatrix} 2 & 0 & 0 & 0 \end{bmatrix}^T$. If we take $g = h$ and $\mathtt{T}(z) = \mathtt{I}_4$, then the matrix identity (4.2) becomes

$$\begin{bmatrix} \mathtt{H}^*(z) + \mathtt{F}(z)(1 - \mathtt{H}(z)\mathtt{H}^*(z)) & \mathtt{I}_4 - \mathtt{F}(z)\mathtt{H}(z) \end{bmatrix} \begin{bmatrix} \mathtt{H}(z) \\ \mathtt{I}_4 - \mathtt{H}^*(z)\mathtt{H}(z) \end{bmatrix} = \mathtt{I}_4 \tag{4.4}$$

where $\mathtt{H}^*(z) = \mathtt{H}(z^{-1})^T$ is the conjugate transpose of $\mathtt{H}(z)$. Hence, from the arguments in the proof of Theorem 5, we obtain a non-redundant wavelet FB. Let $\mathtt{A}(z)$ be its analysis

polyphase matrix. Then the first row of $\mathtt{A}(z)$ is $\mathtt{H}(z)$, and the second through the fourth rows of $\mathtt{A}(z)$ are the transpose of the following column vectors

$$\begin{bmatrix} -\frac{1}{8} - \frac{1}{8}z_1 \\ -\frac{1}{16}z_1^{-1} + \frac{7}{8} - \frac{1}{16}z_1 \\ -\frac{1}{16}z_2^{-1} - \frac{1}{16}z_2^{-1}z_1 - \frac{1}{16} - \frac{1}{16}z_1 \\ -\frac{1}{16}z_1^{-1}z_2^{-1} - \frac{1}{16}z_2^{-1} - \frac{1}{16} - \frac{1}{16}z_1 \end{bmatrix},$$

$$\begin{bmatrix} -\frac{1}{8} - \frac{1}{8}z_2 \\ -\frac{1}{16}z_1^{-1} - \frac{1}{16}z_1^{-1}z_2 - \frac{1}{16} - \frac{1}{16}z_2 \\ -\frac{1}{16}z_2^{-1} + \frac{7}{8} - \frac{1}{16}z_2 \\ -\frac{1}{16}z_1^{-1}z_2^{-1} - \frac{1}{16}z_1^{-1} - \frac{1}{16} - \frac{1}{16}z_2 \end{bmatrix},$$

$$\begin{bmatrix} -\frac{1}{8} - \frac{1}{8}z_1z_2 \\ -\frac{1}{16}z_1^{-1} - \frac{1}{16} - \frac{1}{16}z_2 - \frac{1}{16}z_1z_2 \\ -\frac{1}{16}z_2^{-1} - \frac{1}{16} - \frac{1}{16}z_1 - \frac{1}{16}z_1z_2 \\ -\frac{1}{16}z_1^{-1}z_2^{-1} + \frac{7}{8} - \frac{1}{16}z_1z_2 \end{bmatrix},$$

respectively. Its synthesis polyphase matrix $\mathtt{S}(z)$ is given as

$$\begin{bmatrix} \alpha(z_1,z_2) & -\frac{1}{2}z_1^{-1} - \frac{1}{2} & -\frac{1}{2}z_2^{-1} - \frac{1}{2} & -\frac{1}{2}z_1^{-1}z_2^{-1} - \frac{1}{2} \\ \frac{1}{4} + \frac{1}{4}z_1 & 1 & 0 & 0 \\ \frac{1}{4} + \frac{1}{4}z_2 & 0 & 1 & 0 \\ \frac{1}{4} + \frac{1}{4}z_1z_2 & 0 & 0 & 1 \end{bmatrix}$$

where $\alpha(z_1,z_2) = \frac{1}{2} + 2(\frac{3}{8} - \frac{1}{16}(z_1^{-1} + z_2^{-1} + z_1^{-1}z_2^{-1} + z_1 + z_2 + z_1z_2))$. In particular, the three synthesis highpass filters, say k_1, k_2, and k_3, are directional and aligned along the nonzero

cosets, i.e.,

$$k_1: \quad \boxed{-\tfrac{1}{2}} \quad 1 \quad -\tfrac{1}{2} \qquad\qquad k_2: \quad \begin{matrix} & -\frac{1}{2} \\ 1 & \\ \boxed{-\frac{1}{2}} & \end{matrix} \qquad\qquad k_3: \quad \begin{matrix} & & -\frac{1}{2} \\ & 1 & \\ \boxed{-\frac{1}{2}} & & \end{matrix}$$

■

Example 1 provides a simple way to construct a 2-D non-redundant wavelet FB from an interpolatory lowpass filter h, and the resulting synthesis highpass filters are directional and very sparse. Since h is interpolatory, other existing methods (e.g. methods in [4,44]) may be used under appropriate choice of parameters in order to give a similar result. In the next example, we show how our method can be used to construct a non-redundant wavelet FB from a non-interpolatory lowpass filter h.

Example 2 (2-D wavelet FB generated from a non-interpolatory lowpass filter). Let h be the lowpass filter associated with the bivariate box spline $B_{1,1,2}$ based on the four directions $(1,0), (0,1)$, $(1,1)$ and $(1,1)$, i.e.

$$h: \quad \begin{matrix} & & \frac{1}{8} & \frac{1}{8} \\ & \frac{1}{4} & \frac{3}{8} & \frac{1}{8} \\ \frac{1}{8} & \boxed{\frac{3}{8}} & \frac{1}{4} & \\ \frac{1}{8} & \frac{1}{8} & & \end{matrix}$$

Then the filter h is no longer interpolatory and its polyphase representation $\mathtt{H}(z)$ is

$$\begin{bmatrix} \frac{3}{8} + \frac{1}{8}z_1^{-1}z_2^{-1} & \frac{1}{8} + \frac{1}{4}z_1^{-1} + \frac{1}{8}z_1^{-1}z_2^{-1} & \frac{1}{8} + \frac{1}{4}z_2^{-1} + \frac{1}{8}z_1^{-1}z_2^{-1} & \frac{1}{8} + \frac{3}{8}z_1^{-1}z_2^{-1} \end{bmatrix}.$$

We choose $\mathtt{F}(z) = \begin{bmatrix} 3 & 0 & 0 & -1 \end{bmatrix}^T$ as a dual of $\mathtt{H}(z)$. As we did in Example 1, we take $g = h$ and $\mathtt{T}(z) = \mathtt{I}_4$. Then we obtain the same identity as in (4.4) of Example 1 for our new $\mathtt{F}(z)$ in this example. By using the arguments in the proof of Theorem 5 again, we obtain

a non-redundant wavelet FB. Let $\mathtt{A}(z)$ be its analysis polyphase matrix. Then the first row of $\mathtt{A}(z)$ is $\mathtt{H}(z)$, whereas the second through the fourth rows of $\mathtt{A}(z)$ are the transpose of the following column vectors

$$\begin{bmatrix} -\frac{1}{16} - \frac{3}{32}z_1 - \frac{1}{32}z_2^{-1} - \frac{3}{64}z_1z_2 - \frac{1}{64}z_1^{-1}z_2^{-1} \\ \frac{29}{32} - \frac{1}{32}z_1 - \frac{1}{32}z_1^{-1} - \frac{1}{32}z_2 - \frac{1}{32}z_2^{-1} - \frac{1}{64}z_1z_2 - \frac{1}{64}z_1^{-1}z_2^{-1} \\ -\frac{1}{32} - \frac{1}{16}z_1 - \frac{1}{16}z_2^{-1} - \frac{1}{64}z_1z_2 - \frac{1}{64}z_1^{-1}z_2^{-1} - \frac{1}{16}z_1z_2^{-1} \\ -\frac{1}{16} - \frac{1}{32}z_1 - \frac{3}{32}z_2^{-1} - \frac{1}{64}z_1z_2 - \frac{3}{64}z_1^{-1}z_2^{-1} \end{bmatrix},$$

$$\begin{bmatrix} -\frac{1}{16} - \frac{1}{32}z_1^{-1} - \frac{3}{32}z_2 - \frac{3}{64}z_1z_2 - \frac{1}{64}z_1^{-1}z_2^{-1} \\ -\frac{1}{32} - \frac{1}{16}z_1^{-1} - \frac{1}{16}z_2 - \frac{1}{64}z_1z_2 - \frac{1}{64}z_1^{-1}z_2^{-1} - \frac{1}{16}z_1^{-1}z_2 \\ \frac{29}{32} - \frac{1}{32}z_1 - \frac{1}{32}z_1^{-1} - \frac{1}{32}z_2 - \frac{1}{32}z_2^{-1} - \frac{1}{64}z_1z_2 - \frac{1}{64}z_1^{-1}z_2^{-1} \\ -\frac{1}{16} - \frac{3}{32}z_1^{-1} - \frac{1}{32}z_2 - \frac{1}{64}z_1z_2 - \frac{3}{64}z_1^{-1}z_2^{-1} \end{bmatrix},$$

$$\begin{bmatrix} \frac{3}{16} - \frac{5}{32}z_1z_2 - \frac{1}{32}z_1^{-1}z_2^{-1} \\ -\frac{1}{12} - \frac{1}{16}z_1^{-1} - \frac{5}{48}z_2 - \frac{5}{96}z_1z_2 - \frac{1}{32}z_1^{-1}z_2^{-1} \\ -\frac{1}{12} - \frac{5}{48}z_1 - \frac{1}{16}z_2^{-1} - \frac{5}{96}z_1z_2 - \frac{1}{32}z_1^{-1}z_2^{-1} \\ \frac{13}{16} - \frac{5}{96}z_1z_2 - \frac{3}{32}z_1^{-1}z_2^{-1} \end{bmatrix},$$

respectively. Its synthesis polyphase matrix $\mathtt{S}(z)$ is given as

$$\mathtt{D(z)} \begin{bmatrix} -\frac{3}{8} - \frac{3}{4}z_1^{-1} - \frac{3}{8}z_1^{-1}z_2^{-1} & -\frac{3}{8} - \frac{3}{4}z_2^{-1} - \frac{3}{8}z_1^{-1}z_2^{-1} & -\frac{3}{8} - \frac{9}{8}z_1^{-1}z_2^{-1} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{8} + \frac{1}{4}z_1^{-1} + \frac{1}{8}z_1^{-1}z_2^{-1} & \frac{1}{8} + \frac{1}{4}z_2^{-1} + \frac{1}{8}z_1^{-1}z_2^{-1} & \frac{9}{8} + \frac{3}{8}z_1^{-1}z_2^{-1} \end{bmatrix}$$

where

$$\mathtt{D}(z) = \begin{bmatrix} \frac{3}{8} + \frac{1}{8}z_1z_2 + 3(\frac{1}{2} - \frac{1}{16}(z_1^{-1} + z_2^{-1} + 2z_1^{-1}z_2^{-1} + z_1 + z_2 + 2z_1z_2)) \\ \frac{1}{8} + \frac{1}{4}z_1 + \frac{1}{8}z_1z_2 \\ \frac{1}{8} + \frac{1}{4}z_2 + \frac{1}{8}z_1z_2 \\ \frac{1}{8} + \frac{3}{8}z_1z_2 - (\frac{1}{2} - \frac{1}{16}(z_1^{-1} + z_2^{-1} + 2z_1^{-1}z_2^{-1} + z_1 + z_2 + 2z_1z_2)) \end{bmatrix}. \quad \blacksquare$$

Below we list two corollaries of Theorem 5, whose proofs are placed in Appendix §4.5. The first corollary says that the accuracy number of the synthesis lowpass filter of the non-redundant wavelet FB in Theorem 5 can be stated in terms of the accuracy number and the flatness number of the other filters involved in the construction. Here, the *flatness number* of a filter f is defined to be the number of zeros of $\sqrt{q} - F(e^{i\omega})$ at $\omega = 0$. Notice that f is a lowpass filter if and only if its flatness number is positive.

Corollary 3 *Let h be a lowpass filter with flatness β_h. Suppose that h has a dual lowpass filter. Let f be a dual lowpass filter of h with accuracy α_f, and let g be a lowpass filter with accuracy α_g and flatness β_g. Suppose that the accuracy number α_g is positive. Then there exists a dual lowpass filter d of h such that the filter d is determined entirely from f, g, and h, and that the accuracy of the filter d is at least* $\min\{\alpha_g, \alpha_f + \beta_g, \alpha_f + \beta_h\}$. ■

In the above corollary, the dual filter d has positive accuracy since $\min\{\alpha_g, \alpha_f + \beta_g, \alpha_f + \beta_h\}$ is clearly positive, which in turn is implied by the positivity of α_g, β_g, and β_h. However, $\min\{\alpha_g, \alpha_f + \beta_g, \alpha_f + \beta_h\}$ may be lagging behind α_h, the accuracy number of the lowpass filter h. In such a case, one may want to find a dual whose accuracy number is at least α_h. The next corollary says that such a dual can always be found.

Corollary 4 *Let h be a lowpass filter with positive accuracy α_h. Suppose that h has a dual lowpass filter f. Then there exists a dual lowpass filter d of h such that the filter d is determined entirely from f and h, and that the accuracy of the filter d is at least α_h.* ■

As we observed in the previous subsection, a new method developed in [44] provides a motivation for our construction method presented in this paper. Indeed, the fact

that it is a special case of our general construction can be shown as follows. We recall the polyphase representation of an interpolatory analysis lowpass filter is given as $\mathtt{H}(z) = [\frac{1}{\sqrt{q}}, H_{\nu_1}(z), \cdots, H_{\nu_{q-1}}(z)]$. Thus we can set $\mathtt{F}(z) = [\sqrt{q}, 0, \cdots, 0]^T$ and $\mathtt{T}(z) = \mathtt{I}_q$ (cf. Example 1). Therefore, in this case, no change of basis is needed and the usual polyphase representation is sufficient. The identity (4.2) in this case becomes

$$\begin{bmatrix} & 0 & -\sqrt{q}H_{\nu_1}(z) & \cdots & -\sqrt{q}H_{\nu_{q-1}}(z) \\ & 0 & & & \\ \mathtt{D}(z) & & & & \\ & \vdots & & \mathtt{I}_{q-1} & \\ & 0 & & & \end{bmatrix} \begin{bmatrix} \mathtt{H}(z) \\ \mathtt{I}_q - \mathtt{G}(z)\mathtt{H}(z) \end{bmatrix} = \mathtt{I}_q,$$

where $\mathtt{D}(z) := \mathtt{G}(z) + \mathtt{F}(z)(1 - \mathtt{H}(z)\mathtt{G}(z))$. By deleting the second column of the first matrix and the second row of the second matrix, we obtain the non-redundant wavelet FB in [44]. Hence our result here can be considered as a generalization of the method in the aforementioned paper.

4.3.3 Algorithms for constructing multi-D wavelet FBs from a single lowpass filter

Our methodology in the previous subsection is very general. In particular, the filters f, g, and h in Corollary 3 or 4 do not, in general, uniquely determine the highpass filters of the associated wavelet FB, which may not be desirable for some applications. The following corollary provides a way to obtain unique highpass filters given f, g, and h by choosing the matrix $\mathtt{T}(z)$ in the proof of Theorem 5 to be a special form. Its proof is placed in Appendix §4.5.

Corollary 5 *Let h be a lowpass filter with accuracy α_h and flatness β_h. Suppose that h has a dual lowpass filter. Let f be a dual lowpass filter of h with accuracy α_f, and let g be a lowpass filter with accuracy α_g and flatness β_g. Suppose that the accuracy numbers α_h and α_g are positive. Let $\mathtt{K}(z)$ be an invertible $q \times q$ matrix such that $\mathtt{K}(z)\mathtt{H}(z)^T = [1, 0, \cdots, 0]^T$ where $\mathtt{H}(z)$ (as a row vector) is the polyphase representation of h. Let d be the filter whose polyphase representation is $\mathtt{G}(z) + \mathtt{F}(z)(1 - \mathtt{H}(z)\mathtt{G}(z))$ where $\mathtt{G}(z)$ and $\mathtt{F}(z)$ are the polyphase representation (as a column vector) of g and f. Let $k_1, \ldots, k_{q-1}$ and $j_1, \ldots, j_{q-1}$ be the filters whose polyphase representations are the 2nd through the qth column of $\mathtt{K}(z)^T$ and the 2nd through the qth row of $[\mathtt{K}(z)^T]^{-1}[\mathtt{I}_q - \mathtt{F}(z)\mathtt{H}(z)][\mathtt{I}_q - \mathtt{G}(z)\mathtt{H}(z)]$, respectively. Then $\{h, j_1, \ldots, j_{q-1}\}$, $\{d, k_1, \ldots, k_{q-1}\}$ form a wavelet FB with at least $\min\{\alpha_h, \alpha_g, \alpha_f + \beta_g, \alpha_f + \beta_h\}$ vanishing moments.* ■

The above corollary provides an algorithm to construct a non-redundant wavelet FB just from a single lowpass filter h, provided that h has positive accuracy and its polyphase representation $\mathtt{H}(z)$ is unimodular. We note that this positive accuracy condition on h and the unimodularity condition on $\mathtt{H}(z)$ are necessary conditions for any lowpass filter to be used for wavelet FBs. In this sense, one can say that our algorithms below work under the minimum assumptions on the lowpass filter h.

Algorithm 1: An algorithm for constructing a non-redundant wavelet FB from a lowpass filter.

Input: *h: a lowpass filter with positive accuracy and with unimodular polyphase representation.*

Outputs: *d: a dual lowpass filter of h with positive accuracy.*

$j_1, \ldots, j_{q-1}$, $k_1, \ldots, k_{q-1}$: highpass filters that form a wavelet FB, together with h and d.

Step 1: *Choose a lowpass filter g with positive accuracy.*

Step 2: *Find a lowpass filter f that is dual to h.*

Step 3: *Find an invertible $q \times q$ matrix $\mathtt{K}(z)$ such that $\mathtt{K}(z)\mathtt{H}(z)^T = [1, 0, \cdots, 0]^T$, where $\mathtt{H}(z)$ (as a row vector) is the polyphase representation of h.*

Step 4: *Set d to be the filter whose polyphase representation is $\mathtt{G}(z) + \mathtt{F}(z)(1 - \mathtt{H}(z)\mathtt{G}(z))$ where $\mathtt{G}(z)$ and $\mathtt{F}(z)$ are the polyphase representation (as a column vector) of g and f.*

Step 5: *Set $k_1, \ldots, k_{q-1}$ to be the filters whose polyphase representations are the 2nd through the qth column vectors of $\mathtt{K}(z)^T$.*

Step 6: *Set $j_1, \ldots, j_{q-1}$ to be the filters whose polyphase representations are the 2nd through the qth row vectors of the matrix $[\mathtt{K}(z)^T]^{-1}[\mathtt{I}_q - \mathtt{F}(z)\mathtt{H}(z)][\mathtt{I}_q - \mathtt{G}(z)\mathtt{H}(z)]$.* ■

The above algorithm starts from a given lowpass filter h to build a wavelet FB, whose analysis lowpass filter is h. The filter g in Step 1 is an arbitrary lowpass filter with positive accuracy. One possible choice is to take $g := h$ as we did in our examples in the previous subsection. The existence of f in Step 2 and $\mathtt{K}(z)$ in Step 3 is due to the facts that h has positive accuracy and $\mathtt{H}(z)$ is unimodular. In fact, one can always choose f to be the filter whose polyphase representation is the first column vector of $\mathtt{K}(z)^T$ once $\mathtt{K}(z)$ is determined. Although algorithms for finding f and $\mathtt{K}(z)$ are implemented in many mathematical softwares such as Maple, Singular and CoCoA, the *QuillenSuslin* package in Maple (cf. Section §4.2.2) is the only implementation that we know to give a square matrix $\mathtt{K}(z)$ for any unimodular $\mathtt{H}(z)$. Given h, once specific f, g and $\mathtt{K}(z)$ are chosen, the wavelet

FB having h as its analysis lowpass filter is uniquely determined.

From Corollary 5, we see that the vanishing moments of the FBs constructed following Algorithm 1 are at least $\min\{\alpha_h, \alpha_g, \alpha_f + \beta_g, \alpha_f + \beta_h\}$. Although this number is clearly positive, which is enough for the FB to be a wavelet FB, it can be lagging behind α_h. By combining Corollary 5 (or Algorithm 1) with the idea used in Corollary 4, one can obtain the following algorithm that provides wavelet FBs whose vanishing moments are at least α_h.

Algorithm 2: An algorithm for constructing a non-redundant wavelet FB from a lowpass filter so that its vanishing moments are at least as many as the accuracy number of the lowpass filter.

Input: *h: a lowpass filter with positive accuracy α_h and with unimodular polyphase representation.*

Outputs: *Ite: the number of iterations performed.*

d: a dual lowpass filter of h with positive accuracy.

$j_1, \ldots, j_{q-1}$, $k_1, \ldots, k_{q-1}$: highpass filters that form, together with h and d, a wavelet FB with at least α_h vanishing moments.

Step 1: *Set $Ite := 1$ and $g := h$.*

Step 2: *Find a lowpass filter f that is dual to h.*

Step 3: *Find an invertible $q \times q$ matrix $\mathtt{K}(z)$ such that $\mathtt{K}(z)\mathtt{H}(z)^T = [1, 0, \cdots, 0]^T$ where $\mathtt{H}(z)$ (as a row vector) is the polyphase representation of h.*

Step 4: *Set d to be the filter whose polyphase representation is $\mathtt{G}(z) + \mathtt{F}(z)(1 - \mathtt{H}(z)\mathtt{G}(z))$ where $\mathtt{G}(z)$ and $\mathtt{F}(z)$ are the polyphase representation (as a column vector) of g and f.*

Step 5: *If* $\alpha_f + (Ite)\beta_h < \alpha_h$, *set* $Ite := Ite + 1$ *and repeat Step 4 with* $f := d$. *Otherwise, go to Step 6.*

Step 6: *Set* $k_1, \ldots, k_{q-1}$ *to be the filters whose polyphase representations are the* $2nd$ *through the* qth *column vectors of* $\mathtt{K}(z)^T$.

Step 7: *Set* $j_1, \ldots, j_{q-1}$ *to be the filters whose polyphase representations are the* $2nd$ *through the* qth *row vectors of the matrix* $[\mathtt{K}(z)^T]^{-1}[\mathtt{I}_q - \mathtt{F}(z)\mathtt{H}(z)][\mathtt{I}_q - \mathtt{G}(z)\mathtt{H}(z)]$. ■

4.4 Summary and outlook

In this paper we presented a new algebraic approach for constructing wavelet FBs using Quillen-Suslin Theorem for Laurent polynomials. Our method is motivated by some existing techniques that were used mostly only for interpolatory filters (cf. Section §4.3.1). Quillen-Suslin Theorem for Laurent polynomials is used to transform the filters in polyphase representation to a special form of valid polyphase representations, for which the existing matrix analysis tools can be readily applied (cf. Section §4.3.2). Our method works for any dimension and for any dilation, but it would be most beneficial for multi-D case since this is where the construction gets more difficult. The method provides algorithms for constructing multi-D wavelet FBs from a single lowpass filter with minimal assumptions: positive accuracy and unimodularity of the polyphase representation (cf. Section §4.3.3).

Our findings in this paper show that constructing multi-D wavelet FBs using the Quillen-Suslin Theorem, a well-known result in Algebraic Geometry, offers some noteworthy advantages over other more traditional approaches. We plan to explore the opportunities to study other challenges in multi-D wavelet FB construction using Algebraic Geometry

techniques in our future researches.

4.5 Appendix

4.5.1 Proof of Corollary 3 in section §4.3.2

We first recall that a filter f has accuracy number $k \in \mathbb{N}$ if and only if its Fourier transform $F(e^{i\omega})$ satisfies

$$F(e^{i(\omega+\gamma)}) = O(|\omega|^k), \quad (\text{near } \omega = 0),$$

for all $\gamma \in \Gamma^* \backslash \{0\}$, and it has flatness $k \in \mathbb{N}$ if and only if

$$\sqrt{q} - F(e^{i\omega}) = O(|\omega|^k), \quad (\text{near } \omega = 0).$$

From the proof of Theorem 5, we know that for any lowpass filters h, f, and g that satisfy the assumptions of Corollary 3, there exists a dual lowpass filter d of h whose polyphase representation satisfies

$$\mathtt{D}(z) = \mathtt{G}(z) + \mathtt{F}(z)(1 - \mathtt{H}(z)\mathtt{G}(z)).$$

The z-transform of d is obtained via

$$\begin{aligned} D(z) = v(z)^*\mathtt{D}(z^\Lambda) &= v(z)^*\mathtt{G}(z^\Lambda) + v(z)^*\mathtt{F}(z^\Lambda)(1 - \mathtt{H}(z^\Lambda)\mathtt{G}(z^\Lambda)) \\ &= G(z) + F(z)(1 - \mathtt{H}(z^\Lambda)\mathtt{G}(z^\Lambda)) \\ &= G(z) + F(z)B(z^\Lambda) \end{aligned}$$

where $B(z) := 1 - \mathtt{H}(z)\mathtt{G}(z)$. Let $z = e^{i(\omega+\gamma)}$, then

$$D(e^{i(\omega+\gamma)}) = G(e^{i(\omega+\gamma)}) + F(e^{i(\omega+\gamma)})B((e^{i(\omega+\gamma)})^\Lambda).$$

Thus it suffices to show that

$$D(e^{i(\omega+\gamma)}) = O(|\omega|^{\min\{\alpha_g,\alpha_f+\beta_g,\alpha_f+\beta_h\}})$$

near $\omega = 0$, for all $\gamma \in \Gamma^* \backslash \{0\}$.

From the fact that $B((e^{i(\omega+\gamma)})^\Lambda) = B((e^{i\omega})^\Lambda)$ for all $\gamma \in \Gamma^* \backslash \{0\}$, and the simple observation (cf. Appendix C in [44])

$$B((e^{i\omega})^\Lambda) = 1 - \frac{1}{q} \sum_{\gamma \in \Gamma^*} H(e^{i(\omega+\gamma)}) G(e^{i(\omega+\gamma)}),$$

we have

$$\begin{aligned} B((e^{i\omega})^\Lambda) =& 1 - \frac{1}{q} H(e^{i\omega}) G(e^{i\omega}) + O(|\omega|^{\alpha_h+\alpha_g}) \\ =& 1 - \frac{1}{q} (\sqrt{q} + O(|\omega|^{\beta_h}))(\sqrt{q} + O(|\omega|^{\beta_g})) + O(|\omega|^{\alpha_h+\alpha_g}) \\ =& O(|\omega|^{\min\{\beta_h,\beta_g,\alpha_h+\alpha_g\}}), \quad (\text{near } \omega = 0). \end{aligned}$$

Therefore

$$\begin{aligned} D(e^{i(\omega+\gamma)}) =& G(e^{i(\omega+\gamma)}) + F(e^{i(\omega+\gamma)}) B((e^{i(\omega+\gamma)})^\Lambda) \\ =& O(|\omega|^{\alpha_g}) + O(|\omega|^{\alpha_f}) O(|\omega|^{\min\{\beta_h,\beta_g,\alpha_h+\alpha_g\}}) \\ =& O(|\omega|^{\min\{\alpha_g,\alpha_f+\beta_g,\alpha_f+\beta_h\}}) \end{aligned}$$

near $\omega = 0$, for all $\gamma \in \Gamma^* \backslash \{0\}$.

4.5.2 Proof of Corollary 4 in section §4.3.2

In this proof, we use an iterative method to construct a dual lowpass filter d of h such that the accuracy number of d is at least α_h. For any lowpass filters h with positive

accuracy α_h, if we let $g := h$ and f be a dual lowpass filter of h, then by Corollary 3 and its proof, we know that there exists a dual lowpass filter d of h whose polyphase representation is

$$\mathtt{D}(z) = \mathtt{G}(z) + \mathtt{F}(z)(1 - \mathtt{H}(z)\mathtt{G}(z)) \tag{4.5}$$

and its accuracy number is at least $\min\{\alpha_g, \alpha_f + \beta_g, \alpha_f + \beta_h\} = \min\{\alpha_h, \alpha_f + \beta_h\}$. If $\alpha_f + \beta_h < \alpha_h$, then we set $f := d$, and use this new f in (4.5) to construct a new d. This new d now has accuracy number at least $\min\{\alpha_h, \alpha_f + 2\beta_h\}$. Since $\beta_h \geq 1$, $\alpha_f + 2\beta_h$ is strictly larger that $\alpha_f + \beta_h$, and if $\alpha_f + 2\beta_h < \alpha_h$, we can iteratively update f to be the new d until $\alpha_f + (Ite)\beta_h \geq \alpha_h$, where Ite denotes the number of iterations. Thus we obtain a dual lowpass filter d whose accuracy number is at least α_h.

4.5.3 Proof of Corollary 5 in section §4.3.3

Since $\mathtt{K}(z)\mathtt{H}(z)^T = [1, 0, \cdots, 0]^T$, we have $\mathtt{H}(z)\mathtt{K}(z)^T = [1, 0, \cdots, 0]$. Therefore, $\mathtt{H}(z) = [1, 0, \cdots, 0][\mathtt{K}(z)^T]^{-1}$, i.e., the first row of $[\mathtt{K}(z)^T]^{-1}$ is $\mathtt{H}(z)$.

Let $\mathtt{T}(z)$ in the proof of Theorem 5 be $[\mathtt{K}(z)^T]^{-1}$. Then $\mathtt{H}^u(z) = \mathtt{H}(z)[\mathtt{T}(z)]^{-1} = [1, 0, \cdots, 0]$ and the first component of $\mathtt{F}^w(z) = \mathtt{T}(z)\mathtt{F}(z)$ is 1 since the first row of $[\mathtt{K}(z)^T]^{-1}$ is $\mathtt{H}(z)$ and f is dual to h.

Then, after some calculation, we see that the identity (4.3) in the proof of Theo-

rem 5 becomes

$$\begin{bmatrix} & 0 & 0 & \cdots & 0 \\ & & & & \\ & 0 & 1 & & \\ \mathtt{D}^w(z) & \vdots & & \ddots & \\ & & & & \\ & 0 & & & 1 \end{bmatrix} \begin{bmatrix} \mathtt{H}^u(z) & & & & \\ & & & & \\ 1 - G^w_{\nu_0}(z) & & 0 & \cdots & 0 \\ -G^w_{\nu_1}(z) - F^w_{\nu_1}(z)(1 - G^w_{\nu_0}(z)) & & 1 & & \\ \vdots & & & \ddots & \\ -G^w_{\nu_{q-1}}(z) - F^w_{\nu_{q-1}}(z)(1 - G^w_{\nu_0}(z)) & & & & 1 \end{bmatrix} = \mathtt{I}_q$$

where $\mathtt{D}^w(z) := \mathtt{G}^w(z) + \mathtt{F}^w(z)(1 - \mathtt{H}^u(z)\mathtt{G}^w(z))$ and $c(z)$ in the reduction matrix $\mathtt{R}(z)$ is taken to be 1.

By deleting the second column of the first matrix and the second row of the second matrix in the above equation, we obtain a non-redundant FB. From Theorem 5, we know that this FB is a wavelet FB. The analysis lowpass filter is h and the synthesis lowpass filter d has polyphase representation $\mathtt{D}(z) = \mathtt{G}(z) + \mathtt{F}(z)(1 - \mathtt{H}(z)\mathtt{G}(z))$. From Corollary 3, we know that the accuracy number of d is at least $\min\{\alpha_g, \alpha_f + \beta_g, \alpha_f + \beta_h\}$. Therefore this wavelet FB has at least $\min\{\alpha_h, \alpha_g, \alpha_f + \beta_g, \alpha_f + \beta_h\}$ vanishing moments.

Let $k_1, \cdots, k_{q-1}$ be the synthesis highpass filters and $j_1, \cdots, j_{q-1}$ be the analysis highpass filters of the non-redundant wavelet FB that we just found. Let $e_0 := [1, 0, \cdots, 0]^T, e_1 = [0, 1, 0, \cdots, 0]^T, \cdots, e_{q-1} = [0, 0, \cdots, 0, 1]^T$ be the standard unit vectors in $\mathbb{R}^q$. Then from the synthesis side (the one derived from the first matrix of the above matrix identity) of the non-redundant wavelet FB, we see that the polyphase representation for the synthesis highpass filter k_i, for $i = 1, \cdots, q-1$, is

$$[\mathtt{T}(z)]^{-1} e_i = \mathtt{K}(z)^T e_i = (i+1)\text{th column of } \mathtt{K}(z)^T.$$

The polyphase representation for the analysis highpass filter j_i, for $i = 1, \cdots, q-1$, can be obtained from the analysis side (the one derived from the second matrix of the above

matrix identity) of the non-redundant wavelet FB. They are

$$
\begin{aligned}
& \Big[\Big(- G^w_{\nu_i}(z) - F^w_{\nu_i}(z)(1 - G^w_{\nu_0}(z))\Big) e_0^T + e_i^T \Big] \mathtt{T}(z) \\
= \quad & \Big[\Big(- e_i^T \mathtt{T}(z)\mathtt{G}(z) - e_i^T \mathtt{T}(z)\mathtt{F}(z)(1 - e_0^T \mathtt{T}(z)\mathtt{G}(z))\Big) e_0^T + e_i^T \Big] \mathtt{T}(z) \\
= \quad & e_i^T \mathtt{T}(z) \Big[- \mathtt{G}(z) e_0^T \mathtt{T}(z) - \mathtt{F}(z) e_0^T \mathtt{T}(z) + \mathtt{F}(z) e_0^T \mathtt{T}(z) \mathtt{G}(z) e_0^T \mathtt{T}(z) + \mathtt{I}_q \Big] \\
= \quad & e_i^T \mathtt{T}(z) \Big[- \mathtt{G}(z)\mathtt{H}(z) - \mathtt{F}(z)\mathtt{H}(z) + \mathtt{F}(z)\mathtt{H}(z)\mathtt{G}(z)\mathtt{H}(z) + \mathtt{I}_q \Big], \quad (\, e_0^T \mathtt{T}(z) = \mathtt{H}(z) \,) \\
= \quad & e_i^T [\mathtt{K}(z)^T]^{-1} [\mathtt{I}_q - \mathtt{F}(z)\mathtt{H}(z)][\mathtt{I}_q - \mathtt{G}(z)\mathtt{H}(z)] \\
= \quad & (i+1)\text{th row of } [\mathtt{K}(z)^T]^{-1} [\mathtt{I}_q - \mathtt{F}(z)\mathtt{H}(z)][\mathtt{I}_q - \mathtt{G}(z)\mathtt{H}(z)], \quad i = 1, \cdots, q-1,
\end{aligned}
$$

and this concludes the proof.

Chapter 5

Appendix: The Design of Non-redundant Directional Wavelet Filter Bank Using 1-D Neville Filters

5.1 Introduction

In the last couple of decades, wavelets have been a popular and useful tool in many applications such as signal and image processing. One of important remaining challenges in wavelets is to construct multi-D directional wavelet systems or wavelet filter banks.

There has been a lot of attempts to develop such wavelet systems or their variants for 2-D or 3-D signals, such as curvelets, contourlets, shearlets, etc. Despite many benefits of

these existing systems, most of them are redundant with possibly huge redundancy factors, and they do not have a trivial generalization to higher dimensions. Although a recent study by the authors provides the construction of non-redundant wavelet filter banks with directional highpass filters for any dimension [4], it only deals with the dyadic dilation matrices. Other approaches based on anisotropic wavelet bases have also been proposed (see, for example, [65, 71, 97] and the references therein). However, these wavelets are designed in continuous domain and implementing them in discrete setting is not trivial.

In this paper, we develop a new method to construct non-redundant wavelet filter banks that can capture the directional information in multi-D signals. Our method is a general designing recipe in the sense that it can work in any dimension for any dilation matrix. In the design, one can even specify the number of directions and which directions to consider.

5.2 Preliminaries

In this section, we review some basic concepts and notations about wavelet filter bank construction. In particular, we review the concept of Neville filters and how to use Neville filters to build multi-D wavelet filter banks.

5.2.1 Notation

In this paper, we use boldface to indicate vectors and matrices. A filter f is a a linear time-invariant operator characterized by its impulse response $\{f(\mathbf{k}) \in \mathbb{R} | \mathbf{k} \in \mathbb{Z}^n\}$.

CHAPTER 5. APPENDIX: THE DESIGN OF NON-REDUNDANT DIRECTIONAL WAVELET FILTER BANK USING 1-D NEVILLE FILTERS

The z-transform of a filter is a Laurent polynomial

$$F(\mathbf{z}) = \sum_{\mathbf{k}} f(\mathbf{k})\mathbf{z}^{-\mathbf{k}}$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ and $\mathbf{z}^{\mathbf{k}} := \prod_{i=1}^{n} z_i^{k_i}$. In this paper, we refer to both the z-transform $F(\mathbf{z})$ and the impulse response $f(\mathbf{k})$ as the filter, and sometimes we omit $\mathbf{z}$ and $\mathbf{k}$ in the parentheses for convenience. Define the *adjoint* of a filter as $[F(\mathbf{z})]^* := F(1/\mathbf{z})$. Throughout this paper, we assume all filters have finite impulse response.

A dilation matrix $\mathbf{D}$ is a $n \times n$ integer matrix with $|\det \mathbf{D}| := m > 1$. Given a dilation matrix $\mathbf{D}$, the set $\mathbb{Z}^n$ of integer grids can be split into m disjoint subsets

$$\mathbb{Z}^n = \bigcup_{i=0}^{m-1} (\mathbf{D}\mathbb{Z}^n + \mathbf{t}_i), \quad \mathbf{t}_i \in \mathbb{Z}^n$$

where $\mathbf{t}_0 = \mathbf{0}$. We call $\{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{m-1}\}$ as a set of *(nonzero) distinct coset representatives* of the dilation matrix $\mathbf{D}$.

A filter bank (FB) consisting of an analysis bank and a synthesis bank is a set of filters. For a given dilation matrix $\mathbf{D}$, a filter in the analysis bank $\{A_i, i = 0, \ldots, l-1\}$ and a filter in the synthesis bank $\{S_i, i = 0, \ldots, l-1\}$ can be written as the sum of m *polyphase components*

$$A_i(\mathbf{z}) = \sum_{j=0}^{m-1} \mathbf{z}^{\mathbf{t}_j} A_{i,j}(\mathbf{z}^{\mathbf{D}}), \quad a_{i,j}(\mathbf{k}) := a_i(\mathbf{D}\mathbf{k} - \mathbf{t}_j) \tag{5.1}$$

$$S_i(\mathbf{z}) = \sum_{j=0}^{m-1} \mathbf{z}^{-\mathbf{t}_j} S_{i,j}(\mathbf{z}^{\mathbf{D}}), \quad s_{i,j}(\mathbf{k}) := s_i(\mathbf{D}\mathbf{k} + \mathbf{t}_j) \tag{5.2}$$

where $\mathbf{z}^{\mathbf{D}} := (\mathbf{z}^{D_1}, \mathbf{z}^{D_2}, \ldots, \mathbf{z}^{D_d})$, D_i is the ith column vector of $\mathbf{D}$. Then the pair of matrices

$$\mathbf{A}(\mathbf{z}) := [A_{i,j}(\mathbf{z})]_{i=0,\ldots,l-1;j=0,\ldots,m-1}$$

$$\mathbf{S}(\mathbf{z}) := [S_{j,i}(\mathbf{z})]_{j=0,\ldots,m-1;i=0,\ldots,l-1}$$

is called the *polyphase matrix representation* [3] of the FB.

A FB satisfies the *perfect reconstruction* condition if the polyphase matrices satisfy $\mathbf{S}(\mathbf{z})\mathbf{A}(\mathbf{z}) = \mathbf{I}_m$, which can happen only when $l \geq m$. A FB is called *non-redundant* if $l = m$.

In this paper, we are only interested in non-redundant FBs satisfying the perfect reconstruction condition, and we assume there are exactly one lowpass filter A_0 in the analysis bank and one lowpass filter S_0 in the synthesis bank. The rest, $A_1, \ldots, A_{m-1}, S_1, \ldots, S_{m-1}$, are all highpass filters.

We use Π_N to denote the set of all polynomials of total degree less than N. We say a FB has $N \in \mathbb{N}$ *vanishing moments* [36] if, for any highpass filter f in the FB, $(f *' \pi)(\mathbb{Z}^n) = 0, \forall \pi \in \Pi_N$, or equivalently,

$$\textstyle\sum_{\mathbf{k}} f(-\mathbf{k})\mathbf{k}^{\mathbf{n}} = 0, \ \forall \mathbf{n} \in \mathbb{N}_0^n, \ |\mathbf{n}| < N$$

where $\mathbf{n} := (n_1, n_2, \ldots, n_n)$, $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ and $|\mathbf{n}| := n_1 + n_2 + \ldots + n_n$. Here we used $(f *' \pi)(\cdot) := \sum_{\mathbf{k} \in \mathbb{Z}^n} f(\mathbf{k})\pi(\cdot - \mathbf{k})$.

5.2.2 Neville filters and their use in wavelet FB construction

In [26], Kovačević and Sweldens introduce a class of filters called *Neville filters* (Definition 1) and their characterization (Result 1). When applied to a sampled polynomial, they result in the same polynomial but shifted by a shift parameter $\boldsymbol{\tau} \in \mathbb{R}^n$.

Definition 1. A filter f is a Neville filter of order N with shift $\boldsymbol{\tau}$ if $(f *' \pi)(\mathbb{Z}^n) = \pi(\mathbb{Z}^n + \boldsymbol{\tau})$, for any $\pi \in \Pi_N$. ■

Result 1 (Proposition 4 in [26]). A filter f is a Neville filter of order N with shift $\boldsymbol{\tau}$ if

and only if f satisfies

$$\sum_{\mathbf{k}} f(-\mathbf{k})\mathbf{k}^{\mathbf{n}} = \boldsymbol{\tau}^{\mathbf{n}},\ \forall \mathbf{n} \in \mathbb{N}_0^n,\ |\mathbf{n}| < N. \tag{5.3}$$

■

In 1-D case, the construction of Neville filters of order N is straightforward. Once we fix the positions of N filter taps, we obtain a linear system with an $N \times N$ coefficient matrix from (5.3). Since the coefficient matrix in this case is a Vandermonde matrix, it is always solvable. In multi-D case, the solvability of the linear system not only depends on the number of filter taps but also on the geometric shape of the filter. Hence it is more challenging to construct a multi-D Neville filter with a prescribed order and shift. An approach based on an algorithm in [98] to solve this problem is proposed in [26], but it is highly non-trivial to control the shape of the filters using that approach.

Using the property of Neville filters, Kovačević and Sweldens propose a method for constructing wavelet FBs based on lifting scheme [48]. They use two lifting steps: predict (cf. R_i) and update (cf. U_i), as shown in (5.4) and (5.5) to build the wavelet FB with desirable vanishing moments:

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} 1 & U_1 & \cdots & U_{m-1} \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -R_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -R_{m-1} & 0 & \cdots & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 - \sum_{i=1}^{m-1} U_i R_i & U_1 & \cdots & U_{m-1} \\ -R_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -R_{m-1} & 0 & \cdots & 1 \end{bmatrix}\end{aligned} \tag{5.4}$$

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ R_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ R_{m-1} & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 & -U_1 & \cdots & -U_{m-1} \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & -U_1 & \cdots & -U_{m-1} \\ R_1 & 1 - R_1U_1 & \cdots & -R_1U_{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ R_{m-1} & -R_{m-1}U_1 & \cdots & 1 - R_{m-1}U_{m-1} \end{bmatrix}, \tag{5.5}$$

where R_i are called predict filters, U_i are called update filters, and $m = |\det \mathbf{D}|$. More precisely, the following is a variant of the result they prove in [26], written in terms of our terminology.

Result 2. Let $\{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{m-1}\}$ be a set of distinct coset representatives of the $n \times n$ dilation matrix $\mathbf{D}$. For $i = 1, \cdots, m-1$, let R_i be a n-D Neville filter of order N with shift $\boldsymbol{\tau}_i = \mathbf{D}^{-1}\mathbf{t}_i$, and U_i be the filter obtained by multiplying $1/m$ to the adjoint of a n-D Neville filter of order N with shifts $\boldsymbol{\tau}_i$. Then the analysis polyphase matrix constructed as (5.4) and the synthesis polyphase matrix constructed as (5.5) form a wavelet FB with N vanishing moments. ■

This construction works for any dilation matrix $\mathbf{D}$ in any dimension. It uses n-D Neville filters with prescribed orders and shifts to construct n-D wavelet FBs.

5.3 Directional wavelet FB design using 1-D Neville filters

In this section, we introduce a method to design directional wavelet FBs using 1-D Neville filters and the lifting based wavelet construction method reviewed in Section §5.2.2.

Let us first define an operator that maps 1-D filters to n-D filters.

Definition 2. Define the operator that maps a 1-D filter F to a n-D filter $\mathcal{M}_{\mathbf{t}}(F)$ along direction $\mathbf{t} \in \mathbb{Z}^n$ as

$$\mathcal{M}_{\mathbf{t}}(F)(\mathbf{z}) := F(\mathbf{z}^{\mathbf{t}}). \qquad \blacksquare$$

The following simple lemma, which says that the operator $\mathcal{M}_{\mathbf{t}}$ preserves the order of Neville filters is a key ingredient of our directional wavelet FB construction.

Lemma 6 *If F is a 1-D Neville filter of order N with shift $\tau \in \mathbb{R}$, then the n-D filter $\mathcal{M}_{\mathbf{t}}(F)$ is a Neville filter of order N with shift $\tau\mathbf{t}, \mathbf{t} \in \mathbb{Z}^n$.* $\blacksquare$

Proof 8 *Let $G := \mathcal{M}_{\mathbf{t}}(F)$, and let g be the impulse response of G. Then, we have*

$$g(\mathbf{k}) = \begin{cases} f(k), & \text{if } \mathbf{k} = k\mathbf{t} \text{ for some } k \in \mathbb{Z}\ , \\ 0, & \text{for all other } \mathbf{k} \in \mathbb{Z}^n, \end{cases}$$

where f is the impulse response of F. Therefore

$$\begin{aligned} \textstyle\sum_{\mathbf{k}} g(-\mathbf{k})\mathbf{k}^{\mathbf{n}} &= \textstyle\sum_k f(-k)(k\mathbf{t})^{\mathbf{n}} = \sum_k f(-k)k^{|\mathbf{n}|}\mathbf{t}^{\mathbf{n}} \\ &= \tau^{|\mathbf{n}|}\mathbf{t}^{\mathbf{n}} = (\tau\mathbf{t})^{\mathbf{n}}, \end{aligned}$$

for any $\mathbf{n} \in \mathbb{N}_0^n$, $|\mathbf{n}| < N$, where the second last equation holds because F is a 1-D Neville filter of order N with shift τ. Thus G is a d-D Neville filter of order N with shift $\tau\mathbf{t}$. $\blacksquare$

Example 1: Mapping 1-D Neville Filter to 2-D. $F(z) = 1/3z + 2/3$ is an 1-D Neville filter of order 2 with shift $\tau = 1/3$. Then mapping it to 2-D along direction $\mathbf{t} = (1, 1)$ results in $\mathcal{M}_{\mathbf{t}}(F)(\mathbf{z}) = 1/3z_1z_2 + 2/3$. It can be easily checked that $\mathcal{M}_{\mathbf{t}}(F)$ is a Neville filter of order 2 with shift $\tau\mathbf{t} = (1/3, 1/3)$.

$$\begin{array}{cc} \frac{1}{3} & \underline{\frac{2}{3}} \end{array} \quad \longrightarrow \quad \begin{array}{cc} 0 & \underline{\frac{2}{3}} \\ \frac{1}{3} & 0 \end{array}$$

Figure 5.1: Mapping 1-D Neville filter to 2-D. The impulse response of F and $\mathcal{M}_{\mathbf{t}}(F)$ in Example 1. Underlined position is the origin.

Figure 5.1 shows the impulse response of F and $\mathcal{M}_{\mathbf{t}}(F)$. ■

From Example 1, we see that the multi-D Neville filter constructed by the operator $\mathcal{M}_{\mathbf{t}}$ is directional along direction $\mathbf{t}$. We now discuss how to use these directional multi-D Neville filters to construct directional wavelet FB.

Let us first look at a simple case when the dilation matrix $\mathbf{D} = c\mathbf{I}_n$ where $c \in \mathbb{Z},\ c > 1$ and $\mathbf{I}_n$ is the identity matrix. In this case, $\mathbf{D}^{-1} = (1/c)\mathbf{I}_n$. The multi-D Neville filters used to construct predict and update filters in Result 2 need to have shift parameters $\boldsymbol{\tau}_i = \mathbf{D}^{-1}\mathbf{t}_i = (1/c)\mathbf{t}_i$. Therefore, it is possible to construct all these multi-D Neville filters by mapping a single 1-D Neville filter with shift $\tau = 1/c$ but with different directions $\mathbf{t}_i$. In this way, we can avoid constructing multi-D Neville filters directly, which is often difficult to do. Moreover, it can be shown that the highpass filters built on these multi-D Neville filters are also directional.

To generalize this idea to a general dilation matrix $\mathbf{D}$, let us consider the shift parameters $\boldsymbol{\tau}_i = \mathbf{D}^{-1}\mathbf{t}_i$ again. In this case, if we factor out $\tau = 1/m$ as the shift parameter for 1-D Neville filters, then $\boldsymbol{\tau}_i = \tau\tilde{\mathbf{t}}_i$, where $\tilde{\mathbf{t}}_i = m\mathbf{D}^{-1}\mathbf{t}_i \in \mathbb{Z}^n$, hence we can map a single 1-D Neville filter with shift $\tau = 1/m$ along different directions $\tilde{\mathbf{t}}_i$. For example, for dilation

matrix

$$\mathbf{D} = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix} \tag{5.6}$$

a set of distinct coset representatives of $\mathbf{D}$ are $\mathbf{t}_1 = (0,1), \mathbf{t}_2 = (1,1), \mathbf{t}_3 = (0,2), \mathbf{t}_4 = (1,2)$. The shift parameters of Neville filters needed to construct wavelet FB are $\boldsymbol{\tau}_1 = (1/5, 2/5), \boldsymbol{\tau}_2 = (3/5, 1/5), \boldsymbol{\tau}_3 = (2/5, 4/5), \boldsymbol{\tau}_4 = (4/5, 3/5)$. Therefore, we can construct all these multi-D Neville filters by mapping one 1-D Neville filter with shift $1/5$ along directions $\tilde{\mathbf{t}}_1 = (1,2), \tilde{\mathbf{t}}_2 = (3,1), \tilde{\mathbf{t}}_3 = (2,4), \tilde{\mathbf{t}}_4 = (4,3)$.

In fact, we can factor out any $\tau = 1/s$, where $s \in \mathbb{Z}$, as the shift parameter for 1-D Neville filters, as long as $\boldsymbol{\tau}_i = \tau\tilde{\mathbf{t}}_i$ and $\tilde{\mathbf{t}}_i = s\mathbf{D}^{-1}\mathbf{t}_i \in \mathbb{Z}^n$. In the simple case when $\mathbf{D} = c\mathbf{I}_d$, $s := c$ can be chosen, while in other cases such as (5.6), $s := m$ can be chosen. Therefore, we have the following theorem. For a general n-D dilation matrix $\mathbf{D}$ with $|\det \mathbf{D}| = m$, we can construct a directional wavelet FB with analysis highpass filters presenting at most $m-1$ different directions as follows.

Theorem 6 *Let $\{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{m-1}\}$ be a set of distinct coset representatives of $\mathbf{D}$. Let s be an integer such that $s\mathbf{D}^{-1}\mathbf{t}_i \in \mathbb{Z}^n$. For $i = 1, \cdots, m-1$, let P_i and Q_i be the 1-D Neville filters of order N with shift $1/s$. Set $\tilde{\mathbf{t}}_i = s\mathbf{D}^{-1}\mathbf{t}_i$. Let d-D filter $R_i := \mathcal{M}_{\tilde{\mathbf{t}}_i}(P_i)$ and $U_i := (1/m)[\mathcal{M}_{\tilde{\mathbf{t}}_i}(Q_i)]^*$. Then the analysis polyphase matrix given by (5.4) and the synthesis polyphase matrix given by (5.5) form a directional FB with N vanishing moments and the analysis highpass filters are placed along directions $\mathbf{t}_i$.* ■

Proof 9 *Since P_i (resp. Q_i) is a 1-D Neville filter of order N with shift $1/s$, by Lemma 1, $R_i = \mathcal{M}_{\tilde{\mathbf{t}}_i}(P_i)$ (resp. $\mathcal{M}_{\tilde{\mathbf{t}}_i}(Q_i)$) is a d-D Neville filter of order N with shift $(1/s)\tilde{\mathbf{t}}_i =$*

$(1/s)s\mathbf{D}^{-1}\mathbf{t}_i = \mathbf{D}^{-1}\mathbf{t}_i$. *Thus* $U_i = (1/m)[\mathcal{M}_{\tilde{\mathbf{t}}_i}(Q_i)]^*$ *is* $1/m$ *times the adjoint of Neville filter of order* N *with shift* $\mathbf{D}^{-1}\mathbf{t}_i$. *By Result 2, we see that (5.4) and (5.5) form a wavelet FB with* N *vanishing moments.*

To prove the directionality of analysis highpass filters, consider the i*th analysis highpass filter denoted by* A_i. *Since*

$$R_i(\mathbf{z}) = \mathcal{M}_{\tilde{\mathbf{t}}_i}(P_i)(\mathbf{z}) = P(\mathbf{z}^{\tilde{\mathbf{t}}_i}) = P(\mathbf{z}^{s\mathbf{D}^{-1}\mathbf{t}_i}),$$

from (5.1) and (5.4), we see that $A_i(\mathbf{z})$ *is equal to*

$$-R_i(\mathbf{z}^{\mathbf{D}}) + \mathbf{z}^{\mathbf{t}_i} = -P_i(\mathbf{z}^{\mathbf{D}s\mathbf{D}^{-1}\mathbf{t}_i}) + \mathbf{z}^{\mathbf{t}_i} = -P_i(\mathbf{z}^{s\mathbf{t}_i}) + \mathbf{z}^{\mathbf{t}_i}.$$

If we replace $\mathbf{z}^{\mathbf{t}_i}$ *with* z *in the last equation on the right hand side, we get a* 1*-D filter* $-P_i(z^s) + z$. *Thus* A_i *can be understood as the result of taking the* 1*-D filter* $-P_i(z^s) + z$ *and placing it in* n*-D space along direction* $\mathbf{t}_i$. ■

Remark 1. In Theorem 1, a single 1-D Neville filter of order N and shift $1/m$ can be used for all of P_i and Q_i, or different 1-D Neville filters can be used. In fact P_i and Q_i can have different orders if we invoke more generalized version of Result 2 from [26]. In this case, if P_i's order is $\tilde{N}_i$ and Q_i's order is N_i, then the vanishing moments of the FB is given as $\min\{\tilde{N}_1, \ldots, \tilde{N}_{m-1}, N_1, \ldots, N_{m-1}\}$. ■

Remark 2. The analysis highpass filters A_i of the FB in Theorem 1 are placed along directions $\mathbf{t}_i \in \mathbb{Z}^n, i = 1, \ldots, m-1$ (not $\tilde{\mathbf{t}}_i = m\mathbf{D}^{-1}\mathbf{t}_i$). Therefore, by carefully choosing the distinct coset representatives of $\mathbf{D}$, one can custom-design the directions of the filters (cf. Example 2). There are *at most* $m-1$ different directions that can be presented by the analysis highpass filters. ■

In the next example, we illustrate how to use Theorem 1 to construct directional wavelet FB.

Example 2: 2-D Directional Wavelet FB with 2 Vanishing Moments. For dilation matrix $\mathbf{D} = 3\mathbf{I}_2$, since $|\det \mathbf{D}| = 9$, there are $9 - 1 = 8$ distinct coset representatives $\{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_8\}$ that we can choose. We know that the directions of coset representatives are exactly the directions of resulting analysis highpass filters. Here we want to choose directions that divide the 2-D plane as equally as possible. Thus we choose $\mathbf{t}_1 = (1,0), \mathbf{t}_2 = (-1,0), \mathbf{t}_3 = (0,1), \mathbf{t}_4 = (0,-1), \mathbf{t}_5 = (2,1), \mathbf{t}_6 = (1,2), \mathbf{t}_7 = (-2,1), \mathbf{t}_8 = (-1,2)$. Then the resulting analysis highpass filters will present 6 different directions in the 2-D plane: approximately, 0° ($\mathbf{t}_1$, $\mathbf{t}_2$), 30° ($\mathbf{t}_5$), 60° ($\mathbf{t}_6$), 90° ($\mathbf{t}_3$, $\mathbf{t}_4$), 120° ($\mathbf{t}_8$) and 150° ($\mathbf{t}_7$) from the positive x-axis.

Next we pick a single 1-D Neville filter of order 2 with shift 1/3 for all P_i and Q_i: $P_i(z) = Q_i(z) = 1/3z + 2/3$, for $i = 1, \ldots, 8$. Theorem 1 says that if we choose, for each i,

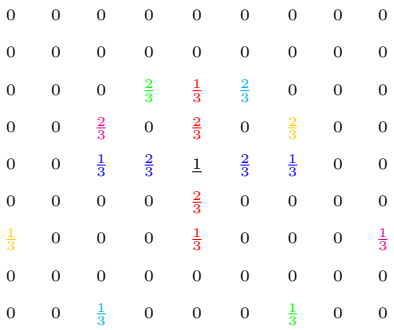
$$\begin{aligned} R_i(\mathbf{z}) &= P_i(\mathbf{z}^{\mathbf{t}_i}) = 1/3\mathbf{z}^{\mathbf{t}_i} + 2/3 \\ U_i(\mathbf{z}) &= (1/m)[Q_i(\mathbf{z}^{\mathbf{t}_i})]^* = (1/9)(1/3\mathbf{z}^{-\mathbf{t}_i} + 2/3) \end{aligned}$$

then we get the wavelet FB with 2 vanishing moments, whose polyphase matrices are $\mathbf{A}$ and $\mathbf{S}$ in (5.4) and (5.5). Using formula (5.1) and (5.2), we can read off the corresponding filters. For example, the resulting synthesis lowpass filter S_0 is

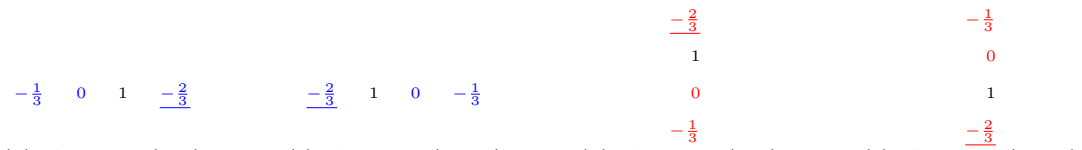
$$S_0(\mathbf{z}) = 1 + \textstyle\sum_{i=1}^{8} \mathbf{z}^{-\mathbf{t}_i} R_i(\mathbf{z}^{\mathbf{D}})$$

and the resulting analysis highpass filter associated with coset representative $\mathbf{t}_5 = (2,1)$ is

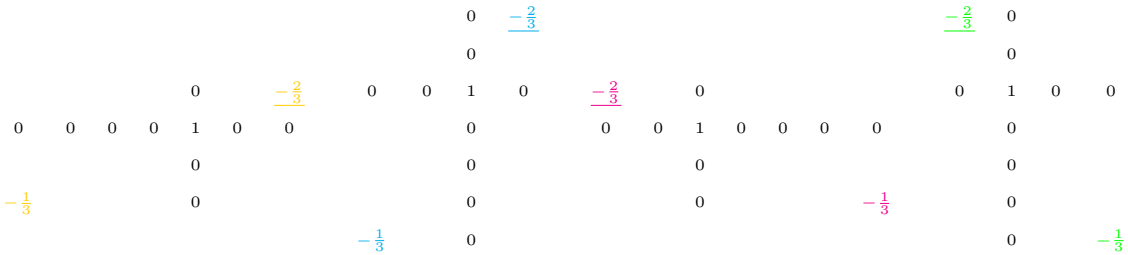
$$A_5(\mathbf{z}) = -R_5(\mathbf{z}^{\mathbf{D}}) + \mathbf{z}^{\mathbf{t}_5} = -(1/3z_1^6 z_2^3 + 2/3) + z_1^2 z_2.$$



(a) Synthesis lowpass filter S_0



(b) $A_1 : \mathbf{t}_1 = (1, 0)$ (c) $A_2 : \mathbf{t}_2 = (-1, 0)$ (d) $A_3 : \mathbf{t}_3 = (0, 1)$ (e) $A_4 : \mathbf{t}_4 = (0, -1)$



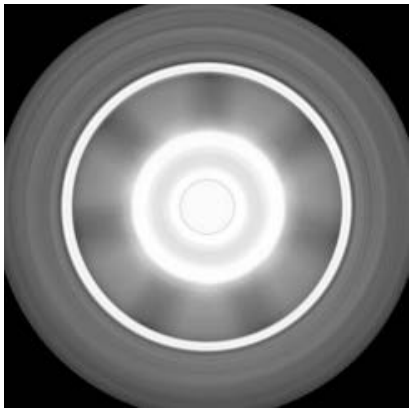
(f) $A_5 : \mathbf{t}_5 = (2, 1)$ (g) $A_6 : \mathbf{t}_6 = (1, 2)$ (h) $A_7 : \mathbf{t}_7 = (-2, 1)$ (i) $A_8 : \mathbf{t}_8 = (-1, 2)$

Figure 5.2: 2-D directional wavelet FB with 2 vanishing moments in Example 2: (a) synthesis lowpass filter, (b)-(i) directional analysis highpass filters with each direction along the coset representatives: $\mathbf{t}_i$, $i = 1, \ldots, 8$.

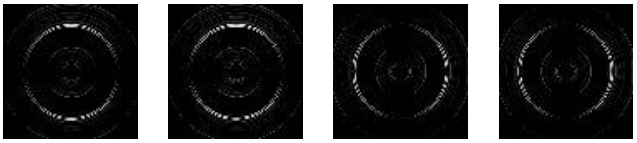
Figure 5.2 shows the synthesis lowpass filter S_0 and the analysis highpass filters A_i, $i = 1, \ldots, 8$. ■

5.4 Experimental result

We did an experiment using the 2-D directional wavelet FB constructed in Example 2. For an original image "circle" (Figure 5.3(a)), we did a 1-level-down decomposition using the analysis highpass filters obtained in Example 2 (as shown in Figure 5.2(b)-(i)). The



(a) Original



(b) A_1 $(1,0)$ (c) A_2 $(-1,0)$ (d) A_3 $(0,1)$ (e) A_4 $(0,-1)$

(f) A_5 $(2,1)$ (g) A_6 $(1,2)$ (h) A_7 $(-2,1)$ (i) A_8 $(-1,2)$

Figure 5.3: (a) The original image "circle", (b)-(i) the images after passing highpass filters $A_1, \ldots, A_8$.

images after passing through each highpass filter (wavelet coefficients) are shown in Figure 5.3(b)-(i). The result shows that different directional components of the circle are captured by different directional highpass filters. A highpass filter with direction $\mathbf{t}$ can mainly capture the directional content that is orthogonal to the direction $\mathbf{t}$.

5.5 Conclusion

In this paper, we developed a method to use 1-D Neville filters to build multi-D directional wavelet FBs. The resulting FB is a non-redundant FB which can capture the directional information in multi-D signals.

Bibliography

[1] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674–693, 1989.

[2] Y. Meyer, *Wavelets and Operators*. Cambridge: Cambridge University Press, 1992.

[3] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley: Wellesley-Cambridge Press, 1997.

[4] Y. Hur and F. Zheng, "Coset Sum: an alternative to the tensor product in wavelet construction," *IEEE Trans. Inform. Theory*, vol. 59, pp. 3554–3571, 2013.

[5] ——, "The design of non-redundant directional wavelet filter bank using 1-D Neville filters," in *Proceedings of 10th International Conference on Sampling Theory and Applications (SampTA)*, 2013, pp. 216–219.

[6] S. D. Riemenschneider and Z. Shen, "Construction of compactly supported biorthogonal wavelets in $L_2(\mathbb{R}^d)$ II," in *Wavelet Applications Signal and Image Processing VII*, 1999, pp. 264–272.

[7] J. Kovačević and M. Vetterli, "Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for $\mathbb{R}^n$," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 533–555, 1992.

[8] A. Cohen and J.-M. Schlenker, "Compactly supported bidimensional wavelet bases with hexagonal symmetry," *Constr. Approx.*, vol. 9, no. 2-3, pp. 209–236, 1993.

[9] J. Kovačević and M. Vetterli, "Nonseparable two- and three-dimensional wavelets," *IEEE Trans. Signal Processing*, vol. 43, no. 5, pp. 1269–1273, 1995.

[10] C. A. Micchelli and Y. Xu, "Reconstruction and decomposition algorithms for biorthogonal multiwavelets," *Multidimensional Systems and Signal Processing*, vol. 8, pp. 31–69, 1997.

[11] S. D. Riemenschneider and Z. Shen, "Multidimensional interpolatory subdivision schemes," *SIAM J. Numer. Anal.*, vol. 34, pp. 2357–2381, 1997.

[12] W. He and M.-J. Lai, "Construction of bivariate compactly supported biorthogonal box spline wavelets with arbitrarily high regularities," *Appl. Comput. Harmon. Anal.*, vol. 6, pp. 53–74, 1999.

[13] H. Ji, S. D. Riemenschneider, and Z. Shen, "Multivariate compactly supported fundamental refinable functions, duals and biorthogonal wavelets," *Stud. Appl. Math.*, vol. 102, pp. 173–204, 1999.

[14] M. Salvatori and P. M. Soardi, "Multivariate compactly supported biorthogonal spline wavelets," *Annali di Matematica*, vol. 181, no. 2, pp. 161–179, 2002.

[15] Q. Chen, C. A. Micchelli, S. Peng, and Y. Xu, "Multivariate filter banks having matrix factorizations," *SIAM J. Matrix Anal. Appl.*, vol. 25, pp. 517–531, 2003.

[16] Q. Chen, C. A. Micchelli, and Y. Xu, "Biorthogonal multivariate filter banks from centrally symmetric matrices," *Linear Algebra and its Applications*, vol. 402, pp. 111–125, 2005.

[17] Z. He, X. You, and Y. Yuan, "Texture image retrieval based on non-tensor product wavelet filter banks," *Signal Process.*, vol. 89, pp. 1501–1510, 2009.

[18] G. Quellec, M. Lamard, G. Cazuguel, B. Cochener, and C. Roux, "Adaptive nonseparable wavelet transform via lifting and its application to content-based image retrieval," *IEEE Trans. Image Processing*, vol. 19, no. 1, pp. 25–35, 2010.

[19] X. You, L. Du, Y. Cheung, and Q. Chen, "A blind watermarking scheme using new nontensor product wavelet filter banks," *IEEE Trans. Image Processing*, vol. 19, no. 12, pp. 3271–3284, 2010.

[20] Z. Zhang, "A new method of constructions of non-tensor product wavelets," *Acta Appl. Math.*, vol. 111, pp. 153–169, 2010.

[21] D. B. H. Tay and N. G. Kingsbury, "Flexible design of multidimensional perfect reconstruction FIR 2-band filters using transformations of variables," *IEEE Trans. Image Processing*, vol. 2, no. 4, pp. 466–480, 1993.

[22] S.-M. Phoong, C. W. Kim, P. P. Vaidyanathan, and R. Ansari, "A new class of two-channel biorthogonal filter banks and wavelet bases," *IEEE Trans. Signal Processing*, vol. 43, no. 3, pp. 649–665, 1995.

[23] R. H. Bamberger and M. J. T. Smith, "A filter bank for the directional decomposition of images: theory and design," *IEEE Trans. Signal Processing*, vol. 40, no. 4, pp. 882–893, 1992.

[24] I. A. Shah and A. A. C. Kalker, "Theory and design of multidimensional QMF sub-band filters from 1-D filters and polynomials using transforms," *Communications, Speech and Vision, IEE Proceedings I*, vol. 140, no. 1, pp. 67–71, 1993.

[25] R. Ansari and C.-L. Lau, "Two-dimensional IIR filters for exact reconstruction in tree-structured sub-band decomposition," *Electronics Letters*, vol. 23, no. 12, pp. 633–634, 1987.

[26] T. A. C. M. Kalker and I. A. Shah, "A group theoretic approach to multidimensional filter banks: theory and applications," *IEEE Trans. Signal Processing*, vol. 44, no. 6, pp. 1392–1405, 1996.

[27] J. M. Shapiro, "Adaptive McClellan transformations for quincunx filter banks," *IEEE Trans. Signal Processing*, vol. 42, no. 3, pp. 642–648, 1994.

[28] Y. Hur and A. Ron, "New constructions of piecewise-constant wavelets," *Electronic Transactions on Numerical Analysis*, vol. 25, pp. 138–157, 2006.

[29] ——, "L-CAMP: Extremely local high-performance wavelet representations in high spatial dimension," *IEEE Trans. Inform. Theory*, vol. 54, pp. 2196–2209, 2008.

[30] Rajan, P. and Reddy, H. and Swamy, M., "Fourfold rotational symmetry in two-dimensional functions," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 30, no. 3, pp. 488–499, 1982.

BIBLIOGRAPHY

[31] A. S. Cavaretta, W. Dahmen, and C. A. Micchelli, *Stationary subdivision.* Memoirs of Amer. Math. Soc., 1991, vol. 93.

[32] B. Han and R.-Q. Jia, "Optimal interpolatory subdivision schemes in multidimensional spaces," *SIAM J. Numer. Anal.*, vol. 36, pp. 105–124, 1999.

[33] A. Ron and Z. Shen, "Affine systems in $L_2(\mathbb{R}^d)$ II: dual systems," *J. Fourier Anal. Appl.*, vol. 3, no. 5, pp. 617–637, 1997.

[34] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 45, no. 5, pp. 485–560, 1992.

[35] Y. Hur and A. Ron, "High-performance very local Riesz wavelet bases of $L_2(\mathbb{R}^n)$," *SIAM Journal on Mathematical Analysis*, vol. 44, pp. 2237–2265, 2012.

[36] D.-R. Chen, B. Han, and S. D. Riemenschneider, "Construction of multivariate biorthogonal wavelets with arbitrary vanishing moments," *Adv. Comput. Math.*, vol. 13, no. 2, pp. 131–165, 2000.

[37] S. G. Mallat, *A Wavelet Tour of Signal Processing.* San Diego, CA: Academic Press, 1999.

[38] B. Han, "Analysis and construction of optimal multivariate biorthogonal wavelets with compact support," *SIAM Journal on Math. Analysis*, vol. 31, pp. 274–304, 2000.

[39] M. Nielsen, "On polynomial symbols for subdivision schemes," *Adv. Comput. Math*, vol. 27, pp. 195–209, 2007.

[40] C. De Boor, K. Höllig, and S. D. Riemenschneider, *Box Splines*. New York, NY: Springer-Verlag, 1993.

[41] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: Soc. Ind. Appl. Math., 1992.

[42] G. Deslauriers and S. Dubuc, "Interpolation dyadique," in *Fractals, Dimensions Non Entières et Applications*, Masson, Paris, 1987, pp. 44–55.

[43] H. Ji and Z. Shen, "Compactly supported (bi)orthogonal wavelets generated by interpolatory refinable functions," *Adv. Comput. Math*, vol. 11, pp. 81–104, 1999.

[44] Y. Hur, "Effortless critical representation of Laplacian pyramid," *IEEE Trans. Signal Processing*, vol. 58, pp. 5584–5596, 2010.

[45] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 532–540, 1983.

[46] W. Lawton, S. L. Lee, and Z. Shen, "An algorithm for matrix extension and wavelet construction," *Math. Comp.*, vol. 65, pp. 723–737, 1996.

[47] H. Park, "Optimal design of synthesis filters in multidimensional perfect reconstruction FIR filter banks using Gröbner bases," *IEEE Trans. Circuits Syst.*, vol. 49, pp. 843–851, 2002.

[48] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, 1996.

[49] J. Kovačević and W. Sweldens, "Wavelet families of increasing order in arbitrary dimensions," *IEEE Trans. Signal Processing*, vol. 9, no. 3, pp. 480–496, 2000.

[50] José Salvador Oliver Gil, "On the design of fast and efficient wavelet image coders with reduced memory usage," Ph.D. dissertation, Universitat Politécnica de Valéncia, 2006.

[51] Wavelet Toolbox, *Version 4.7 (R2011a)*. Natick, Massachusetts: The MathWorks Inc., 2011.

[52] Rabbani, M. and Joshi, R.L. and Jones, P.W., "JPEG 2000 core coding system (part 1)," *The JPEG 2000 Suite*, pp. 1–69, 2009.

[53] E. J. Candès and D. L. Donoho, "Curvelets - a surprisingly effective nonadaptive representation for objects with edges," in *Curve and Surface Fitting: Saint-Malo 1999*, A. Cohen, C. Rabut, and L. L. Schumaker, Eds. Nashville, TN: Vanderbilt University Press, 1999.

[54] E. Candès, L. Demanet, D. Donoho, and L. Ying, "Fast discrete curvelet transforms," *Multiscale modeling and simulation*, vol. 5, no. 3, pp. 861–899, 2006.

[55] CurveLab Toolbox, *Version 2.1.3*. Candès, E. and Demanet, L. and Ying, L., 2008.

[56] Y. Hur and A. Ron, "CAPlets: wavelet representations without wavelets," 2005, preprint. Available: ftp://ftp.cs.wisc.edu/Approx/huron.ps (Original preprint was exceedingly lengthy, and it was resubmitted to a journal after splitting in 2009.).

[57] L. R. et al., *Scientific Visualization: Advances and Challenges*. IEEE Computer Society Press, 1994.

[58] G. Uytterhoeven, "Wavelets: software and applications," Ph.D. dissertation, Dept. Computerwetenschappen, Katholieke Universiteit Leuven, Leuven, Belgium, 1999.

[59] J. D. Villasenor, R. A. Ergas, and P. L. Donoho, "Seismic data compression using high-dimensional wavelet transforms," in *Data Compression Conference, 1996. DCC'96. Proceedings.* IEEE, 1996, pp. 396–405.

[60] C.-H. Hsia, J.-M. Guo, and J.-S. Chiang, "A fast discrete wavelet transform algorithm for visual processing applications," *Signal Processing*, vol. 92, no. 1, pp. 89–106, 2012.

[61] J. Oliver and M. P. Malumbres, "On the design of fast wavelet transform algorithms with low memory requirements," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 237–248, 2008.

[62] H. Meng and Z. Wang, "Fast spatial combinative lifting algorithm of wavelet transform using the 9/7 filter for image block compression," *Electronics Letters*, vol. 36, no. 21, pp. 1766–1767, 2000.

[63] K. K. Parhi and T. Nishitani, "VLSI architectures for discrete wavelet transforms," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 1, no. 2, pp. 191–202, 1993.

[64] Q. Dai, X. Chen, and C. Lin, "A novel VLSI architecture for multidimensional discrete wavelet transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 8, pp. 1105–1110, 2004.

[65] J.-P. Antoine, P. Vandergheynst, and R. Murenzi, "Two-dimensional directional

wavelets in image processing," *International journal of imaging systems and technology*, vol. 7, no. 3, pp. 152–165, 1996.

[66] R. A. DeVore, S. V. Konyagin, and V. N. Temlyakov, "Hyperbolic wavelet approximation," *Constructive Approximation*, vol. 14, no. 1, pp. 1–26, 1998.

[67] E. J. Candès, "Harmonic analysis of neural networks," *Appl. Comput. Harmon. Anal.*, vol. 6, pp. 197–218, 1999.

[68] E. Le Pennec and S. Mallat, "Image compression with geometrical wavelets," in *Proceedings of International Conference on Image Processing*, 2000, pp. 661–664.

[69] M. Vetterli, "Wavelets, approximation, and compression," *IEEE Signal Processing Magazine*, vol. 18(5), pp. 59–73, 2001.

[70] J. L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Processing*, vol. 11(6), pp. 670–684, 2002.

[71] H. Triebel, "Wavelet bases in anisotropic function spaces," in *Proc. Function Spaces, Differential Operators and Nonlinear Analysis (FSDONA2004)*, Milovy, Czech Republic, 2005, pp. 370–387.

[72] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Trans. Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.

[73] S. Pollock and I. L. Cascio, "Non-dyadic wavelet analysis," in *Optimisation, Econometric and Financial Analysis*. Springer, 2007, pp. 167–203.

[74] R. Xiong, J. Xu, and F. Wu, "A lifting-based wavelet transform supporting non-dyadic spatial scalability," in *Image Processing, 2006 IEEE International Conference on*. IEEE, 2006, pp. 1861–1864.

[75] C. Gupta, C. Lakshminarayan, S. Wang, and A. Mehta, "Non-dyadic haar wavelets for streaming and sensor data," in *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*. IEEE, 2010, pp. 569–580.

[76] Y. Hur and F. Zheng, "Designing thin wavelet filters," in *Proceedings of 45th Asilomar Conference on Signals, Systems and Computers*, 2011, pp. 2019–2024.

[77] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[78] M. N. Do and M. Vetterli, "Framing pyramids," *IEEE Trans. Signal Processing*, vol. 51, no. 9, pp. 2329–2342, 2003.

[79] A. K. Soman and P. P. Vaidyanathan, "Generalized polyphase representation and application to coding gain enhancement," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 41, no. 9, pp. 627–630, 1994.

[80] R. G. Swan, "Projective modules over Laurent polynomial rings," *Trans. Amer. Math. Soc*, vol. 237, pp. 111–120, 1978.

[81] J. R. Ragazzini and L. A. Zadeh, "The analysis of sampled-data systems," *Trans. Am. Inst. Elec. Eng*, vol. 71, pp. 225–234, 1952.

[82] J.-P. Serre, "Faisceaux Algébriques Cohérents," *Ann. Math.*, vol. 61, pp. 191–274, 1955.

[83] D. Quillen, "Projective modules over polynomial rings," *Inventiones mathematicae*, vol. 36, no. 1, pp. 167–171, 1976.

[84] A. A. Suslin, "Projective modules over a polynomial ring are free," *Soviet Math Dokl*, vol. 17, no. 4, pp. 1160–1164, 1976.

[85] D. Hilbert, "Ueber die vollen invariantensysteme," *Mathematische Annalen*, vol. 42, no. 3, pp. 313–373, 1893. [Online]. Available: http://dx.doi.org/10.1007/BF01444162

[86] H. Park, "A computational theory of Laurent polynomial rings and multidimensional FIR systems," Ph.D. dissertation, UNIVERSITY of CALIFORNIA, BERKELEY, 1995.

[87] A. Logar and B. Sturmfels, "Algorithms for the Quillen-Suslin theorem," *Journal of Algebra*, vol. 145, pp. 231–239, 1992.

[88] H. Park and C. Woodburn, "An algorithmic proof of suslin's stability theorem for polynomial rings," *Journal of Algebra*, vol. 178, no. 1, pp. 277–298, 1995.

[89] M. Amidou and I. Yengui, "An algorithm for unimodular completion over Laurent polynomial rings," *Linear Algebra and its Applications*, vol. 429, no. 7, pp. 1687 – 1698, 2008.

[90] H. Park, "Symbolic computation and signal processing," *Journal of Symbolic Computation*, vol. 37, no. 2, pp. 209–226, 2004.

[91] R.-Q. Jia, "Approximation properties of multivariate wavelets," *Math. Comp.*, vol. 67, pp. 647–665, 1998.

[92] M. Unser, "An improved least squares Laplacian pyramid for image compression," *Signal Process.*, vol. 27, pp. 187–203, 1992.

[93] S. Toelg and T. Poggio, "Towards an example-based image compression architecture for video-conferencing," 1994, A.I. Memo No. 1494, M.I.T.

[94] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proc. ACM SIGGRAPH*, 1995, pp. 229–238.

[95] P. Shui and Z. Bao, "Construction of nearly orthogonal interpolating wavelets," *Signal processing*, vol. 79, no. 3, pp. 289–300, 1999.

[96] Y. Hur, "Committee Algorithm: an easy way to construct wavelet filters," in *Proc. IEEE ICASSP*, 2012, pp. 3485–3488.

[97] R. A. DeVore, S. V. Konyagin, and V. N. Temlyakov, "Hyperbolic wavelet approximation," *Constructive Approximation*, vol. 14, no. 1, pp. 1–26, 1998.

[98] C. de Boor and A. Ron, "On multivariate polynomial interpolation," *Constructive Approximation*, vol. 6, no. 3, pp. 287–302, 1990.

Vita

Fang Zheng

PERSONAL INFORMATION

Birth: September 23, 1987, Fushun, China

Home: 110 West 39th Street, Baltimore, MD 21210, U.S.

EDUCATION

Johns Hopkins University, Baltimore, MD, U.S.

- Ph.D. in Applied Mathematics and Statistics, September 2010 - March 2014

  Thesis: Algebraic Approaches for Constructing Multi-D Wavelets.

- M.S.E. in Applied Mathematics and Statistics, September 2008 - May 2010

Beijing Forestry University, Beijing, China

- B.S. in Mathematics, September 2004 - June 2008

VITA

TEACHING EXPERIENCE

At Johns Hopkins, I was the instructor for the following course:

- 550.282 A Hands-On Introduction to Matlab, Intersession 2014

At Johns Hopkins, I was the teaching assistant for the following courses:

- 550.361 Introduction to Optimization I
- 550.310 Probability & Statistics for Engieering
- 550.111 Statistical Analysis I

RESEARCH EXPERIENCE

Research Assistant, Johns Hopkins University, Baltimore, MD, U.S.

- Research projects focus on high dimensional wavelet constructions, wavelets with fast algorithms and directional wavelets, June 2010 - May 2014.

Research Assistant, Beijing Forestry University, Beijing, China

- Research projects focus on the application of longitudinal data analysis method to forestry data, November 2007 - May 2008.

PUBLICATIONS

- *The Design of Non-redundant Directional Wavelet Filter Bank Using* 1*-D Neville Filters.* Y. Hur and F. Zheng. Proceedings of SampTA 2013.

VITA

- *Coset Sum: An Alternative to The Tensor Product in Wavelet Construction.* Y. Hur and F. Zheng. IEEE Transactions on Information Theory, Vol. 59, No. 6, June 2013, pp. 3554–3571.

- *Designing Thin Wavelet Filters.* Y. Hur and F. Zheng. Proceedings of 45th Asilomar Conference on Signals, Systems and Computers, 2011.

- *Multi-D Wavelet Filter Bank Design Using Quillen-Suslin Theorem for Laurent Polynomials.* Y. Hur, H. Park and F. Zheng. Submitted for publication, 2014.

- *Prime Coset Sum: A Systematic Method for Designing Multi-D Wavelet Filter Banks with Fast Algorithms.* Y. Hur and F. Zheng. Submitted for publication, 2014.

PRESENTATIONS

- Joint Mathematics Meeting (JMM), Baltimore, MD, January 2014. *The design of non-redundant directional wavelet filter bank using* 1*-D Neville filters*

- 18th Conference of the International Linear Algebra Society (ILAS), Providence, RI, June 2013. *A new algebraic approach to the construction of multidimensional wavelet filter banks.*

- 10th Conference on Sampling Theory and Applications (SampTA), Bremen, Germany, July 2013. *The design of non-redundant directional wavelet filter bank using* 1*-D Neville filters.*

VITA

- International Conference on Wavelets and Applications, St. Petersburg, Russia, July 2012. *Coset sum: an alternative to the tensor product to construct multi-D wavelets.*

- SIAM Conference on Imaging Science, Philadelphia, PA, May 2012. *Compressing aerial images using a critical representation of Laplacian Pyramid.*

- Joint Mathematics Meeting (JMM), Boston, MA, January 2012. *Coset sum: an alternative to the tensor product in wavelet construction.*